

PDP-11/70

CPU DIAGNOSTIC PART 1
MD-11-DEKBA-B

EP-DEKBA-B-DL-A
COPYRIGHT © 1976
FICHE 1 OF 1

NOV 1976
digital
MADE IN USA

The microfiche card contains a grid of 120 frames of diagnostic data for the PDP-11/70 CPU. The frames are arranged in 10 rows and 12 columns. Each frame contains a small, high-contrast image of a diagnostic screen or data table, which is too small to read clearly. The data appears to be organized into columns and rows, possibly representing different test results or system parameters. The overall appearance is that of a standard microfiche card used for data storage and retrieval.

.REM !

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DEK9A-B-0
 PRODUCT NAME: PDP-11/70 CPU DIAGNOSTIC PART 1
 DATE CREATED: 21-AUG-75
 MAINTAINER: DIAGNOSTIC ENGINEERING
 AUTHORS: DON MONROE

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1975 BY DIGITAL EQUIPMENT CORPORATION

03-NOV-75 11:18 PAGE 2
 CP, DIAGNOSTIC PART 1 MAIN 11 27 73
 801

70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95

CONTENTS

1. ABSTRACT
2. REQUIREMENTS
 - 2.1 EQUIPMENT
 - 2.2 STORAGE
 - 2.3 PRELIMINARY PROGRAMS
3. LOADING PROCEDURE
 - 3.1 METHOD
4. STARTING PROCEDURE
 - 4.1 CONTROL SWITCH SETTINGS
 - 4.2 STARTING ADDRESS
 - 4.3 PROGRAM AND OPERATOR ACTION
5. OPERATING PROCEDURE
 - 5.1 OPERATIONAL SWITCH SETTINGS
 - 5.2 SUBROUTINE ABSTRACTS
 - 5.3 OPERATOR ACTION
6. ERRORS
 - 6.1 ERROR HALTS AND DESCRIPTION
 - 6.2 ERROR RECOVERY
7. RESTRICTIONS
 - 7.1 STARTING RESTRICTIONS
 - 7.2 OPERATING RESTRICTIONS
8. MISCELLANEOUS
 - 8.1 EXECUTION TIME
 - 8.2 STACK POINTER
 - 8.3 PASS COUNT
 - 8.4 ITERATIONS
 - 8.5 SPECIAL REGISTERS
 - 8.6 T BIT TRAPPING
 - 8.7 OSCILLOSCOPE SYNC POINTS
 - 8.8 CACHE CONTROL
9. PROGRAM DESCRIPTION
 - 9.1 DEKBA
10. LISTINGS
 - 10.1 DEKBA

100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139

1. ABSTRACT

DEKBA/B ARE PROGRAMS DESIGNED TO DETECT AND REPORT LOGIC FAULTS IN THE PDP 11/70 CENTRAL PROCESSING UNIT. THEY CONSISTS OF 210(8) INDIVIDUAL TESTS CAREFULLY DESIGNED AND SEQUENCED TO DETECT AND ATTEMPT TO IDENTIFY LOGIC FAULTS AT A MINIMUM HARDWARE/SOFTWARE LEVEL. THESE TESTS ARE PARTITIONED INTO TWO STAND-ALONE PROGRAMS AS DESCRIBED BELOW.

A. BASIC INSTRUCTION TESTS

DEKBA CONSISTS OF A LOGICALLY SEQUENCED SET OF INSTRUCTION TESTS DESIGNED TO VERIFY THE INTEGRITY OF THOSE INSTRUCTIONS AND LOGIC OPERATIONS USED BY THE UTILITY ROUTINES THAT PROVIDE ERROR REPORTING AND SCOPE LOOPING FACILITIES FOR DEKBB.

ANY FAULT DETECTED IN THIS PROGRAM CAUSES THE PROGRAM TO "HALT" WITH THE CONSOLE ADDRESS LIGHTS INDICATING THE ERROR PROGRAM COUNTER AND THE CONSOLE DATA LIGHTS SHOWING THE TEST NUMBER (FOR TESTS 24 AND ABOVE). ADDITIONAL FAULT IDENTIFICATION INFORMATION IS AVAILABLE IN THE PROGRAM ANNOTATION FOR THE FAILING TEST.

IF THE PROGRAM HALTS AT LOCATION 6 OR 12 (ADDRESS LIGHTS OF 10 OR 14) THE PROGRAM ANNOTATION FOR THE INDICATED TEST NUMBER, SHOULD GIVE A CLUE TO THE PROBLEM. TO LOOP ON THE ERROR THE HALT MUST BE REPLACED BY THE OCTAL CODE SHOWN IN THE COMMENT FIELD OF THE HALT AND THE PROGRAM RESTARTED AT 200, OR THE START ADDRESS OF THAT PARTICULAR TEST.

B. ADVANCED INSTRUCTION AND MISCELLANEOUS LOGIC TESTS

DEKBB CONSISTS OF A LOGICALLY SEQUENCED SET OF INSTRUCTION TESTS FOLLOWED BY A SET OF MISCELLANEOUS LOGIC TESTS. THE INSTRUCTION TESTS COMPLETE THE TEST OF THE PDP 11/70 INSTRUCTION REPERTOIRE. THE LOGIC TESTS VERIFY SUCH THINGS AS: 1) THE INTERNAL REGISTERS; 2) REGISTERS SET 1; 3) INTERNAL INTERRUPTS; 4) BUS REQUEST LEVELS 4, 5, AND 6; 5) INTERNAL TRAPS AND ABORTS; 6) OUTER MODE SELECTION; AND 7) EXTERNAL TRAPS AND ABORTS. EACH TEST IN THIS PROGRAM CALLS A "SCOPE LOOP" UTILITY THAT FACILITATES USER CONTROL OF TEST SELECTION AND EXECUTION VIA THE CONSOLE SWITCH REGISTER.

UPON DETECTION OF A LOGIC FAULT EACH TEST IN THIS SECTION CALLS AN "ERROR SERVICE" THAT REPORTS IT AS HARD COPY ON THE CONSOLE TERMINAL DEVICE. THE ERROR SERVICE ROUTINE ALSO FACILITATES USER CONTROL OF THE PROGRAM SEQUENCE VIA

140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192

CONSOLE SWITCH REGISTER OPTIONS. AFTER REPORTING THE ERROR THE PROGRAM CONTINUES ON ITS NORMAL SEQUENCE UNLESS MODIFIED BY THE USER ACTIVATING THE "HALT ON ERROR" SWITCH OPTION.

C. IMPORTANT NOTE

THE PROGRAM ANNOTATION IN DEKBA AND THE TYPED ERROR REPORTS IN DEKBB ARE BASED UPON THE KNOWLEDGE THAT ALL PREVIOUS TESTS WERE FAULTLESS AND THAT THERE IS ONLY ONE SINGLE POINT FAILURE IN THE PROCESSOR. THIS MEANS THAT IF EITHER PROGRAM, OR THE PROGRAMS THEMSELVES, ARE NOT RUN IN SEQUENCE, THE ERROR MESSAGE MAY NOT BE VALID.

ALTHOUGH EACH ERROR ANNOTATION AND TYPED MESSAGE CONCLUSION HAS BEEN PROVEN BY PHYSICAL FAULT INSERTION (ONE SIGNAL STUCK LOW), IT IS HUMANLY IMPOSSIBLE TO GUARANTEE THAT THE ERROR REPORT IS 100% CORRECT. THE SOLE FUNCTION OF THE ERROR REPORT IS TO DIRECT THE USER TO THE MOST PROBABLE AREA OF FAILURE.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP 11/70 CPU WITH OPERATORS CONSOLE
LA30 OR EQUIVALENT TERMINAL

2.2 STORAGE

DEKBA REQUIRES 16K TO LOAD AND RUN
DEKBB REQUIRES 16K TO LOAD AND 32K TO RUN

2.3 PRELIMINARY PROGRAMS

DEKBA REQUIRES THAT TWO INSTRUCTIONS WORK:
"BR" AND "HALT"

DEKBB REQUIRES THAT DEKBA RUN

3. LOADING PROCEDURE

3.1 METHOD

BOTH DEKBA AND DEKBB ARE LOADED FROM THE XXDP MEDIA.
REFER TO THE XXDP MANUAL FOR FURTHER INFORMATION.

193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE 5.1

4.2 STARTING ADDRESS

200

4.3 PROGRAM AND OPERATOR ACTION

A. DEKBA

1. LOAD PROGRAM INTO MEMORY (SEE SECTION 3)
2. LOAD ADDRESS 200
3. PRESS START
5. THE PROGRAM WILL LOOP UNTIL THE HALT SWITCH IS PRESSED

B. DEKBB

1. LOAD PROGRAM INTO MEMORY (SEE SECTION 3)
2. ENSURE RH CONTROLLER IS ENABLED, IF SW<5>=0
3. IF AN RKDS IS AVAILABLE (AND THERE WAS NO RH) ENSURE AT LEAST ONE DRIVE IS ENABLED; IF SW<5>=0
4. LOAD ADDRESS 200
5. SET SWITCHES (SEE SECTION 5.1)
6. PRESS START
7. THE PROGRAM WILL LOOP AND AN END OF PASS MESSAGE WILL BE TYPED EVERY PASS.

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

A. DEKBA

NONE

B. DEKBB

WITH SW<15:0>=0 THE PROGRAM WILL PRINT OUT ON ERRORS AND CONTINUE. "END OF PASS" WILL BE TYPED AT THE COMPLETION OF EACH PASS.

244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298

THE SWITCH SETTINGS ARE:

SW<15>=1 ... HALT ON ERROR
 SW<14>=1 ... LOOP ON TEST
 SW<13>=1 ... INHIBIT ERROR TYPEOUTS
 SW<12>=1 ... INHIBIT T BIT TRAPPING
 SW<11>=1 ... INHIBIT ITERATIONS
 SW<10>=1 ... RING BELL ON ERROR
 SW<9> =1 ... LOOP ON ERROR
 SW<8> =1 ... LOOP ON TEST IN SW<7:0>
 SW<7> =1 ... NO ACTION
 SW<6> =1 ... SKIP BUS REQUEST 6 TESTING
 SW<5> =1 ... SKIP BUS REQUEST 5 TESTING
 SW<4> =1 ... SKIP BUS REQUEST 4 TESTING
 SW<3> =1 ... SKIP MEMORY MANAGEMENT TESTING
 SW<2> =1 ... SKIP CACHE TESTING
 SW<1> =1 ... SKIP MAP BOX TESTING
 SW<0> =1 ... SKIP OPERATOR INTERVENTION TESTING

5.2 SUBROUTINE ABSTRACTS

A. DEKBA

SEE 5.2.4 AND 5.2.5

B. DEKBB

5.2.1 SPURIOUS ERROR HANDLER

THIS ROUTINE IS CALLED BY AN UNEXPECTED TRAP TO LOCATION 4 OR 114. IT PRINTS A SHORT MESSAGE FOLLOWED BY THE PROGRAM COUNTER AT THE TIME OF THE TRAP AND THE APPROPRIATE ERROR REGISTER (I.E. THE CPU ERROR REGISTER IN CASES OF A TRAP TO 4 AND THE MEMORY ERROR REGISTER IN CASES OF A TRAP TO 114).

5.2.2 SCOPE

THIS SUBROUTINE CALL (VIA AN IOT INSTRUCTION) IS PLACED BETWEEN EACH TEST. IT RECORDS THE STARTING ADDRESS OF EACH TEST IN LOCATION "\$LPADR" AND "\$LPERR" AS IT IS BEING ENTERED. IT ALSO CONTROLS TEST ITERATION, LOOPING, AND SEQUENTIAL FLOW CHECKS (SEE 5.2.8).

5.2.3 ERROR

THIS SUBROUTINE CALL (VIA A EMT INSTRUCTION) IS USED TO REPORT ALL ERRORS. (REFER TO 6)

5.2.4 TRAP CATCHER

A ".+2" - "HALT" SEQUENCE IS REPEATED FROM LOCATION 0 TO

299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346

LOCATION 776 TO CATCH ANY UNEXPECTED TRAPS (EXCEPT 4 AND 114). THUS, ANY UNEXPECTED TRAPS WILL HALT AT THE DEVICE TRAP VECTOR +2.

5.2.5 TRAP

A NUMBER OF SUBROUTINES ARE CALLED BY THE TRAP INSTRUCTION. FOLLOWING ARE THE CALLS USED AND THE LABEL OF THE STARTING ADDRESS OF THE SUBROUTINES.

5.2.5.1 TYPE (\$TYPE)

ROUTINE TO TYPE AN ASCII STRING ON THE TTY.

THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.

5.2.5.2 TYPEOC (\$TYPOC)

ROUTINE TO CONVERT A 16-BIT BINARY NUMBER TO A 6-DIGIT OCTAL NUMBER AND TYPE IT.

5.2.5.3 TYPEDS (\$TYPDS)

ROUTINE TO CONVERT A 16-BIT BINARY NUMBER TO A 5-DIGIT SIGNED DECIMAL NUMBER AND TYPE IT.

5.2.6 POWER DOWN AND UP

THIS SUBROUTINE CALL (VIA A POWER DOWN) SAVES THE RETURN PC UPON A POWER DOWN. WHEN POWER IS RESTORED A MESSAGE IS TYPED AND THE TEST WILL RESTART.

5.2.7 MONITOR RESTORE (QUIT)

THIS SUBROUTINE IS ENTERED BY TYPING A "CONTROL C" OR WHEN THE END OF PASS IS REACHED AND THE PROGRAM IS RUNNING UNDER A MONITOR. SEE 7.2 FOR DETAILS.

5.2.8 CHECK TEST SEQUENCE (SEQENC)

THIS ROUTINE IS CALLED IN THE SCOPE ROUTINE. IT COMPARES THE ADDRESS OF THE SCOPE CALL WITH THE ADDRESS POINTED TO BY \$STNM IN THE "TEST ADDRESS TABLE". IF THEY DON'T COMPARE, A MESSAGE IS TYPED, INDICATING THAT A TEST WAS SKIPPED.

347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402

5.2.9 PERIPHERAL DETERMINATOR AND INTERRUPT ENABLE ROUTINES

5.2.9.1 PERIPHERAL DETERMINATOR

THIS ROUTINE IS EXECUTED, IN LINE, BETWEEN TESTS 32 AND 33 OF DEKBB. IT CHECKS THE SYSTEM TO DETERMINE IF ONE OF FOUR PERIPHERALS IS AVAILABLE (RS04, RPO4, TM, OR RK05) FOR BUS REQUEST LEVEL 5 TESTING AND IF A LINE CLOCK IS AVAILABLE FOR LEVEL 6 TESTING. IF A DEVICE IS FOUND, THE ADDRESS OF "INTSSU" (INTERRUPT 5 SUBROUTINE) AND \$KW11/P (LINE CLOCK INTERRUPT SUBROUTINE) IS PLACED IN LOCATIONS "INTERS" AND "INTER6" RESPECTIVELY.

5.2.9.2 INTERRUPT ENABLE ROUTINES

THESE ROUTINES ARE CALLED VIA A "JSR PC,@INTERX" (X=5 OR 6). THE ROUTINE SETS UP AND RESPONDS TO A BUS REQUEST. IF THE BR DOES NOT WORK THE RETURN PC IS INCREMENTED BY 2 AND THE RETURN IS MADE.

.MAIN. MACY11 27(657) 13-MAR-75 10:08 PAGE 8
DEKBA.TMP

5.3 OPERATOR ACTION

THE LAST TEST OF DEKBB REQUIRES OPERATOR INTERVENTION. THIS TEST IS ONLY EXECUTED ON PASS 1, IF SW<0>=0. QUESTIONS ARE TYPED ON THE TELETYPE AND THE OPERATOR MUST RESPOND EITHER ON THE CONSOLE OR ON THE TELETYPE.

IF LOCATION 42 IS NON-ZERO, INDICATING THAT THE PROGRAM WAS LOADED BY A MONITOR, THIS TEST IS SKIPPED ON ALL PASSES.

6. ERRORS

6.1 ERROR HALTS AND DESCRIPTION

A. DEKBA

EVERY ERROR IN DEKBA HALTS THE PROCESSOR. THE COMMENT FIELD OF THE HALT INSTRUCTION CONTAINS THE NAME OF THE SIGNAL THAT WAS MOST LIKELY TO HAVE CAUSED THE ERROR. ALSO, IN THE COMMENT FIELD, IS THE OCTAL CODE THAT SHOULD REPLACE THE HALT IF LOOP ON ERROR IS DESIRED. IF THE PROGRAM HALTS AT LOCATION 6 OR 12 THE USER SHOULD LOOK IN THE TEST DESCRIPTION, OF THE TEST THAT FAILED, TO FIND THE MOST LIKELY CAUSE OF THE ERROR.

B. DEKBB

NONE OF THE ERRORS IN DEKBB HALT THE PROCESSOR IF SW<15>=0. THERE ARE OVER 450(8) UNIQUE ERRORS THAT CAN OCCUR IN THIS

J01

403
404
405
406

PROGRAM. WHEN AN ERROR IS ENCOUNTERED THE CALL TO THE
ERROR ROUTINE IS MADE AND IF SW<13> IS NOT SET, AN ERROR
MESSAGE PERTAINING TO THE ERROR WILL BE TYPED. EACH ERROR
TYPE OUT WILL CONTAIN THE FOLLOWING:

K01

407
408
409
410
411

1. AN ERROR MESSAGE
2. A DATA HEADER
3. A DATA STRING

412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462

THE DATA STRING WILL CONTAIN, AT A MINIMUM, THE ERROR PC AND THE TEST NUMBER. IN SOME CASES THE EXPECTED AND ACTUAL VALUES OF A REGISTER ARE ALSO INCLUDED.

REFER TO THE LISTING UNDER \$ERRTB FOR THE TYPES OF ERRORS THAT CAN OCCUR.

6.2 ERROR RECOVERY

A. DEKBA

ERROR RECOVERY IS STRICTLY BY USER INTERVENTION.

B. DEKBB

SW<15:9>=0 - MOST ERRORS WILL CAUSE EXECUTION TO GO TO THE START OF THE NEXT TEST AFTER THE MESSAGE IS TYPED. A FEW TESTS ARE DIVIDED INTO SECTIONS. IN THESE TESTS AN ERROR WILL CAUSE EXECUTION TO GO TO THE NEXT SECTION.

SW<15>=1 - PRESSING THE CONSOLE CONTINUE WILL CAUSE THE PROGRAM TO TYPE AN ERROR MESSAGE AND HALT. PRESSING THE CONSOLE CONTINUE AGAIN WILL CAUSE THE PROGRAM TO CONTINUE AS IF SW<15>=0.

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS

A. DEKBA

NONE

B. DEKBB

IF THE USER WANTS TO RUN THE BUS REQUEST 5 TEST HE MUST ENSURE THAT EITHER AN RH CONTROLLER IS ACTIVE OR THAT A UNIBUS DEVICE (RK, RS, RP, TM) IS ACTIVE.

7.2 OPERATING RESTRICTIONS

A. DEKBA

NONE

463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514

B. DEKBB

SINCE THE PROGRAM COULD POSSIBLY DESTROY A MONITOR, IN PAGE 6, ALL LOCATIONS BETWEEN 152000 AND 157776 ARE SAVED AT THE BOTTOM OF THE PROGRAM. TO RESTORE THESE LOCATIONS A "CONTROL C" SHOULD BE TYPED ON THE TERMINAL. THE LOCATIONS WILL BE RESTORED, A MESSAGE TYPED, AND THE PROCESSOR WILL HALT.

IF THE PROGRAM IS RUNNING UNDER A MONITOR THE LOCATIONS ARE RESTORED AND CONTROL IS RETURNED TO THE MONITOR THRU THE END OF PASS LINKAGE.

8. MISCELLANEOUS

8.1 EXECUTION TIME

A. DEKBA

FIVE(5) SECONDS PER END OF PASS MESSAGE IF RUNNING UNDER A MONITOR. 4 MINUTES IF THE PROGRAM WAS DUMPED.

B. DEKBB

THE FIRST PASS TAKES APPROXIMATELY 8 SECONDS. ALL SUBSEQUENT PASSES TAKE APPROXIMATELY 3 MINUTES.

8.2 STACK POINTER

STACK IS INITIALLY SET TO 1100.

8.3 PASS COUNT

A PROGRAM PASS THRU COUNT IS KEPT IN "\$PASS".

A. DEKBA

IF THE PROGRAM IS RUNNING UNDER A MONITOR OR WAS LOADED BY ACT 11 THE PROGRAM MAKES 144(8) PASSES FOR EACH END OF PASS MESSAGE. IF THE PROGRAM WAS DUMPED, 10000(8) PASSES ARE MADE FOR EACH END OF PASS MESSAGE. THE PASS COUNT IS DISPLAYED IN THE DATA LIGHTS.

B. DEKBB

THE PROGRAM MAKES 1 PASS FOR EACH END OF PASS MESSAGE.

515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557

8.4 ITERATIONS

A. DEKBA

NONE

B. DEKBB

THE FIRST PASS OF THE PROGRAM WILL AUTOMATICALLY INHIBIT ITERATIONS. ALL SUBSEQUENT PASSES WILL PERFORM FULL, (2000 DECIMAL) ITERATIONS.

8.5 SPECIAL REGISTERS

A. DEKBA

R0 IS RESERVED FOR THE TEST NUMBER.

B. DEKBB

NONE

8.6 T BIT TRAPPING

A. DEKBA

NONE

B. DEKBB

EVERY OTHER PASS, STARTING WITH PASS 2, RUNS WITH THE T BIT ON. THIS CAUSES EVERY INSTRUCTION TO T BIT TRAP THEREFORE, IT IS NOT POSSIBLE TO "SINGLE INSTRUCTION" THE TEST WITHOUT TURNING THE T BIT OFF.

CERTAIN TESTS AUTOMATTICALLY TURN IT OFF IF IT WAS ON. THESE TESTS WILL ALSO TURN IT BACK ON UNLESS THE FOLLOWING TEST REQUIRES THAT IT ALSO BE OFF.

5000
5001
5002
5003
5004
5005
5006
5007
5008
5009
5010
5011
5012
5013
5014
5015
5016
5017
5018
5019
5020
5021
5022
5023
5024
5025
5026
5027
5028
5029
5030
5031
5032
5033
5034
5035
5036
5037
5038
5039
5040
5041
5042
5043
5044
5045
5046
5047
5048
5049
5050
5051
5052
5053
5054
5055
5056
5057
5058
5059
5060
5061
5062
5063
5064
5065
5066
5067
5068
5069
5070
5071
5072
5073
5074
5075
5076
5077
5078
5079
5080
5081
5082
5083
5084
5085
5086
5087
5088
5089
5090
5091
5092
5093
5094
5095
5096
5097
5098
5099
5100

8.7 OSCILLOSCOPE SYNC POINTS

A. DEKBA

BEGINNING WITH TEST 24 EACH TEST HAS AN OSCILLOSCOPE SYNC INSTRUCTION. THE ADDRESS OF THE CONDITION CODE ROM STATE (44) IS IN THE PROCESSOR MICROBREAK REGISTER (ADDRESS 17777770). THIS WILL CAUSE PIN AE1 (SLOT 10) ON THE BACKPLANE TO GO HIGH EACH TIME A CONDITION CODE (OR NCP) INSTRUCTION IS EXECUTED. THEREFORE, IF THE OSCILLOSCOPE EXTERNAL SYNC IS CONNECTED TO THIS PIN AND THE SYNC SELECT PUT ON EXTERNAL THE OSCILLOSCOPE WILL BE SYNCHRONIZED WITH THE INSTRUCTION IMMEDIATELY PRECEDING THE INSTRUCTION UNDER TEST (IUT).

B. DEKBB

ONLY TESTS 1 THRU 20 CONTAIN SYNC INSTRUCTIONS.

8.8 CACHE CONTROL

THE FIRST PASS OF BOTH PROGRAMS RUN WITH THE CACHE DISABLED (FORCING MISSES IN BOTH GROUPS). ALL SUBSEQUENT PASSES RUN WITH THE CACH ENABLED.

C02

PDP 11 70 CPU DIAGNOSTIC PART 1

DECDOC VER 00.0

603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629

DOCUMENT

PDP 11/70 CPU DIAGNOSTIC PART 1

COPYRIGHT 1975
DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASS. 01754

A

TABLE OF CONTENTS

630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679

28	BASIC DEFINITIONS
153	CACHE REGISTER DEFINITIONS
164	CPU REGISTER DEFINITIONS
178	MEMORY MANAGEMENT DEFINITIONS
327	UNIBUS MAP REGISTER DEFINITIONS
419	TRAP CATCHER
426	STARTING ADDRESS(ES)
432	ACT11 HOOKS
458	COMMON TAGS
506	ERROR POINTER TABLE
1509	
1559	
1616	
2311	
2681	
2950	
3259	
3460	
3543	
3815	END OF PASS ROUTINE
3851	TYPE ROUTINE
3924	BINARY TO OCTAL (ASCII) AND TYPE

680
681
682
683
684
685
686
687
688
689
690
691

TABLE OF CONTENTS

4002	CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
4070	TRAP DECODER
4085	TRAP TABLE

692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743

17 COPYRIGHT (C) JULY 21, 1975
DIGITAL EQUIPMENT CORP.
MAYNARD, MASS. 01754

PROGRAM BY DONALD W. MONROE

THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
PACKAGE (MAINDEC-11-DZQAC-A5).

28	***** BASIC DEFINITIONS *****
30	INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
44	MISCELLANEOUS DEFINITIONS
50	GENERAL PURPOSE REGISTER DEFINITIONS
71	PRIORITY LEVEL DEFINITIONS
81	"SWITCH REGISTER" SWITCH DEFINITIONS
109	DATA BIT DEFINITIONS (BIT00 TO BIT15)
137	BASIC "CPU" TRAP VECTOR ADDRESSES
153	***** CACHE REGISTER DEFINITIONS *****
164	***** CPU REGISTER DEFINITIONS *****
178	***** MEMORY MANAGEMENT DEFINITIONS *****
181	MEMORY MANAGEMENT STATUS REGISTER ADDRESSES
192	USER "I" PAGE DESCRIPTOR REGISTERS
203	USER "D" PAGE DESCRIPTOR REGISTORS
214	USER "I" PAGE ADDRESS REGISTERS
225	USER "D" PAGE ADDRESS REGISTERS

PDP 11/70 CPU DIAGNOSTIC PART 1

DECDOC VER 00.0

744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799

236 SUPERVISOR "I" PAGE DESCRIPTOR REGISTERS

247 SUPERVISOR "D" PAGE DESCRIPTOR REGISTERS

258 SUPERVISOR "I" PAGE ADDRESS REGISTERS

269 SUPERVISOR "D" PAGE ADDRESS REGISTERS

280 KERNEL "I" PAGE DESCRIPTOR REGISTERS

291 KERNEL "D" PAGE DESCRIPTOR REGISTERS

302 KERNEL "I" PAGE ADDRESS REGISTERS

313 KERNEL "D" PAGE ADDRESS REGISTERS

327 *****
UNIBUS MAP REGISTER DEFINITIONS

330 THE LOWER 16 BITS OF THE MAP REGISTERS ARE LABELED 'MAPLXX'
THE UPPER 6 BITS OF THE MAP REGISTERS ARE LABELED 'MAPHXX'

419 *****
TRAP CATCHER

422 ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS

426 *****
STARTING ADDRESS(ES)

432 *****
ACT11 HOOKS

434 THE FOLLOWING LOCATIONS ARE SETUP TO BE USED WITH ACT11

LOCATION 46 WILL CONTAIN THE ADDRESS OF THE LOGICAL
END OF THE PROGRAM.
LOCATION 52 IS USED TO SPECIFY PROGRAM OPERATING REQUIREMENTS
AND/OR RESTRICTIONS. THIS IS ACCOMPLISHED BY SETTING VARIOUS BIT
TO A ONE OR A ZERO. THE BITS USED AND THERE MEANING ARE:

BIT 15=1 PROGRAM SHOULD BE POWER FAILED WHILE RUNNING
=0 NO POWER FAIL DESIRED

BIT 14=1 PROGRAM RUN TIME IS MEMORY SIZE DEPENDENT
=0 RUN TIME IS NOT MEMORY SIZE DEPENDENT

BITS 13-0 MUST BE ZERO'S

H02

800

PDP 11/70 CPU DIAGNOSTIC PART 1

DECDOC VER 00.0

801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856

458 *****
COMMON TAGS

460 THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
USED IN THE PROGRAM.

506 *****
ERROR POINTER TABLE

508 THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCU
THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
LOCATION \$ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE I
NOTE1: IF \$ITEMB IS 0 THE ONLY PERTINENT DATA IS (\$ERRPC).
NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS

514 EM ;:POINTS TO THE ERROR MESSAGE
DH ;:POINTS TO THE DATA HEADER
DT ;:POINTS TO THE DATA
DF ;:POINTS TO THE DATA FORMAT

522 FOLLOWING IS AN INDEX OF THE POSSIBLE FAILURES THAT
CAUSE A TRAP TO LOCATIONS 4, 10, 14, OR 24 OR THAT
CAUSE THE PROCESSOR TO HANG (H). NGT=NOT GETTING THRU

TEST NUMBER	EFFECT	CAUSE
25	10	RACH NEG.B*DMD N
25	24	RACH A2 RAB00 NO
25	H	RACH E59(6) NOT OR NGT RACL RADR
26	H	RACL RADROG NOT
26	4	EITHER IRCC SM35 OR RACL 270 BAD
26	14	IRCC CO R4903 NO
26	4	RACL RADRO3 INPU SRC CONST ADDED
27	H	RACK RADRO1 NOT
31	10	RACK AO RAB01 NO
31	4	NGT RACL RADRO1 RACK BRCAB05 NOT NGT PACL RADRO5
31	4	IS ON TOP OF STA DST CONST ADDED
32	10	RACE AO RAB00 DO
33	10	RACE AO RAB02 DO
34	10	RACE E44 IS BAD
34	10	IRCB K/CLASS STU
34	H	GRAB OBD(1) STUC

J02

857

Line	Address	Hex	Text
			DECD0C VER 00.0
858			
859			
860	34	H	GRAB DRMX00 STUC
861			
862	35	10	RACE BIN*SMD H D
863	35	10	IR DECODE ROM W0
864			
865	558		
866	36	10	RACE BIN*SMD FAI
867	36	10	IR DECODE ROM W0
868			
869	37	10	RACE E45 BAD
870			
871	40	10	RACE AD RAB00 D0
872	40	10	IRCB(JMP+JSR) IS
873	40	H	IRCB FJ CLASS IS
874			
875	41	10	RACE E45 IS BAD
876			
877	42	10	RACE E33 IS BAD
878			
879	43	10	RACH U/CLASS NOT
880			SS-2 ON TOP OF S
881	43	10	EITHER RACE E10
882			RACE BIN NOT GOI
883			SS ON TOP OF STA
884			
885	44	10	RACJ AFIR 14(1)
886	44	10	IRCB E38(6) STUC
887			
888	45	H	GRAB OBD(0) STUC
889			
890	46	10	RACE E33 BAD
891			
892	54	10	RACF E3 DOES NOT
893			
894	56	10	PART PCLASS FIEL
895			
896	60	10	PART PCLASS FIEL
897			
898	62	10	IRCC C0 RAB03 ST
899			OR FORK C MUX IN
900			
901	65	10	B FORK MUX SELEC
902	65	H	IRCB B0 RAB04 NG
903	65	H	B FORK MUX INPUT
904			IRCC B0 RAB00 ST
905	65	4	B FORK MUX INPUT
906			IRCB B0 RAB00 ST
907	65	H	IRCB E46(10) STU
908			
909	67	10	RACE E35(1) BAD
910			
911	70	10	B FORK MUX STROB
912			
913	75	10	RACE JMP+JSR+SWA

L02

DEKBAB.P11

914

PDP 11/70 CPU DIAGNOSTIC PART 1

DECDOC VER 00.0

75 10

IRCB E63 BAD-R5

105 4
105 4RACF (HALT:OP CO
RACE E7 BAD-ODC

613

THE FOLLOWING FIVE CONDITION CODE AND BRANCH TESTS ARE A
FUNCTION OF RACK F TRUE 1. SECTION 1 OF EACH TEST IS
DEPENDENT ON TRUE 1 NOT GOING HIGH, WHILE SECTION 2 IS
DEPENDENT ON TRUE ONE GOING HIGH.

620 TEST 1 CCC*BRANCH THRU FET.13

THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TR

SECTION 1		
BCS	E58(13,12)	L,H
BMI	E59(10,11,9)	H,L,H
BVS	E48(5,3,4)	H,L,H
BLOS	E59(4,3,5)	H,L,H

642 TEST 2 SEC*BRANCH THRU FET.13 AND FET.11

THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TR

SECTION 1		
BMI	E58(13,12)	H,L
BVS	E58(13,12)	H,L
SECTION 2		
BCC	E58(13,12)	H,H

667 TEST 3 SEV*BRANCH THRU FET.13 AND FET.11

THE FOLLOWING IS A LIST OF PATTERNS PUT ON THE GATES OF

SECTION 1		
BCS	E48(5,3,4)	L,H,H
BMI	E48(5,3,4)	H,H,L
SECTION 2		
BVC	E48(5,3,4)	H,H,H

692 TEST 4 SEZ*BRANCH THRU FET.13 AND FET.11

THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TR

SECTION 1		
BCS	E59(4,3,5)	H,H,L
BMI	E59(4,3,5)	L,H,H
SECTION 2		
BHI	E59(4,3,5)	H,H,H

717 TEST 5 SEN*BRANCH THRU FET.13 AND FET.11

THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TR

915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968

PDP 11/70 CPU DIAGNOSTIC PART 1

DECDOC VER 00.0

969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019

720

SECTION 1		
BLOS	E59(10,11,9)	H,H,L
BVS	E59(10,11,9)	L,H,H
SECTION 2		
BPL	E59(10,11,9)	H,H,H

742

THE FOLLOWING SEVEN TESTS ARE A FUNCTION OF RACF TRUE 2. SECTION 1 OF EACH TEST IS DEPENDENT ON TRUE 2 NOT GOING HIGH WHILE SECTION 2 IS DEPENDENT ON TRUE 2 GOING HIGH.

TEST 6 BRANCHES THRU FET.13

THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TR

SECTION 1		
BLE	E58(2,1)H,L	E59(13,1,2)L,H,H
BLT	E59(13,1,2)L,H,H	E48(1,13,2)H,H,L
BEQ	E58(2,1)H,L	E58(10,9)H,L

764

TEST 7 BRANCH THRU FET.13 AND FET.12

THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TR

SECTION 1		
BEQ	E48(1,13,2)	L,H,H
SECTION 2		
BGT	E48(1,13,2)	H,H,H

788

TEST 10 BRANCH THRU FET.13 AND FET.12

THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TR

SECTION 1		
BLT	E58(2,1)	L,H
SECTION 2		
BNE	E58(2,1)	H,H

811

TEST 11 BRANCH THRU FET.13 AND FET.12

THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TR

SECTION 1		
BEQ	E59(13,1,2)	H,H,L
SECTION 2		
BGE	E59(13,1,2)	H,H,H

834

TEST 12 BRANCHES THRU FET.13 AND FET.12

THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TR

SECTION 1		
BEQ	E58(2,1)H,L	E58(10,9)H,L
BLT	E59(13,1,2)H,L,H	E48(1,13,2)H,L,H

1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073

850 TEST 13 UNIARY AND BINARY (SMD)

THE FOLLOWING TEST TESTS ALL THE E/CLASS INSTRUCTIONS WITH A DESTINATION MODE OF 0 AND DESTINATION FIELD OF NOT 7. THIS CLASS CONSISTS OF ALL THE UNIARY (EXCEPT NEG) INSTRUCTIONS AND ALL THE BINARY INSTRUCTIONS WITH A SOURCE MODE OF 0.

1143 TEST 14 REGISTER SELECTION TEST

THIS TEST ENSURES THAT THE 6 ADDRESS LINES INTO THE GENERAL PURPOSE REGISTERS (GPR) ARE NOT STUCK. THE LABELS OF THE ADDRESS LINES ARE:
GSAX GENERAL SOURCE ADDRESS LINE
GDAX GENERAL DESTINATION ADDRESS LINE
WHERE X STANDS FOR LINE 0, 1, OR 2.
THE CLASSES OF ERRORS DESCRIBED IN THIS TEST

1152 ARE DEFINED AS FOLLOWS:

- CLASS A=GDAX OK
- GSAX STUCK
- CLASS B=GSAX OK
- GDAX STUCK
- CLASS C=GSAX STUCK
- GDAX STUCK

1248 TEST 15 GPR1 STUCK BIT TEST

LOADS GPR1 WITH ZEROS AND ONES AND COMPARES R1 SOURCE AN DESTINATIONS WITH R0. IF THE COMPARISON FAILS A BIT IS S

1274 TEST 16 GPR2 STUCK BIT TEST

LOADS GPR2 WITH ZEROS AND ONES AND COMPARES R2 SOURCE AN DESTINATION WITH R0.

1300 TEST 17 GPR3 STUCK BIT TEST

LOADS GPR3 WITH ZEROS AND ONES AND COMPARES R3 SOURCE AND DESTINATION WITH R0.

1326 TEST 20 GPR4 STUCK BIT TEST

LOADS GPR4 WITH ZEROS AND ONES AND COMPARES R4 SOURCE AND DESTINATION WITH R0.

1352 TEST 21 GPR5 STUCK BIT TEST

LOADS R5 WITH ZEROS AND ONES AND COMPARES R5 SOURCE AND DESTINATION WITH R0.

1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117

1378 TEST 22 GPR6 STUCK BIT TEST

LOADS R6 WITH ZEROS AND ONES AND COMPARES
R6 SOURCE AND DESTINATION WITH R0.

1404 TEST 23 GPR SHORTED BIT TEST

TEST IF GPR'S 1 THRU 6 HAVE TWO BITS TIED TOGETHER

NOTE: R0 IS CONSIDERED "HARDCORE"

1509

1511 TEST 24 ONE MICROSTATE (E/CLASS*DMD*DF7)

THIS TEST EXECUTES AN ADD INSTRUCTION WITH SMD,DMD, AND
IF THE TEST FAILS THE SAME FLOW (EXC. 90) IS
TRIED WITH A PENDING BRQ (T BIT TRAP) INSTEAD OF A DF7.
A FORK A FAILURE IS REPORTED. IF IT PASSES, A TEST 1 FAI

ROM FLOW-30

1559

1561 TEST 25 TWO MICROSTATES (NEG*DMD)

THIS TEST EXECUTES A NEGATE INSTRUCTION WITH DMD.

IF FORK A FAILS EXECUTION WOULD GO TO EITHER RSD.00, ZAP
OR FOP.00.

FOP.00 WILL CAUSE THE PROCESSOR TO HANG.

RSD.00 WOULD CAUSE A TRAP TO LOCATION 10. THIS WOULD ONL
IF RACH NEG.B*DMD DID NOT GO HIGH.

ZAP.00 WOULD CAUSE A TRAP TO LOCATION 24. THIS WOULD ONL
IF RACH A2 RAB00 DID NOT GO LOW.

ROM FLOW-301,210

1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173

1616

1618 TEST 26 THREE MICROSTATES (BIN*SM1*DMO*-DF7*SRO(0))

IF FORK A FAILS EXECUTION WILL GO TO EITHER EXEC.80 OR D
EXC.80 WILL HANG THE PROCESSOR IN THE PAUSE STATE AT MIC
THIS WILL ONLY HAPPEN IF RA CL RADROO IS NOT GOING LOW DU
TO RACF A1 RAB00 (AFIR59(1))*[-BIN+SM01]*U/CLASS).
D12.00 WOULD MOV THE PC TO LOCATION 0.

IF FORK C FAILS EXECUTION WOULD GO FROM S13.10 TO D00.80
D45.01 OR S13.20 OR D12.00 OR JSR.10 OR ASC.80 OR RTI.50
D00.80 WOULD SWAP THE BYTES OF THE SOURCE OPERAND BEFORE
PUTTING THEM IN R5.
D45.01 WOULD MOVE THE PC TO LOCATION 0.
S13.20 WILL EXECUTE A SM3 (AND NO AUTO INC) INSTR. THIS
AN ODD ADDRESS TRAP SINCE LOCATION POSERR CONTAINS AN 00
JSR.10 WOULD PUSH THE ADDR OF POSERR ONTO THE STACK.
ASC.80 WILL HALT AT 8\$.
RTI.50 WILL CAUSE 1004XX TO BE PLACED IN THE PS WORD
AND THE PROCESSOR WILL TRAP TO LOCATION 14.
FOP.50 WILL ?????.

1638

ASH.20 WOULD CAUSE A BAD CC.
IF THE SRC CONST FAILS IN STATE S13.00 AND ADDS 1 OR 3,
ADDRESS TRAP WILL OCCUR.
IF THE SRC CONSTANT ADDS 2, THE ERROR AT 6\$ WILL REPORT

ROM FLOW-21,27,205

1700 TEST 27 THREE MICROSTATES (BIN*SM2*DMO*-DF7*SRO(0))

THIS TEST IS THE SAME AS THE PREVIOUS TEST EXCEPT FOR TH
SM. A WORD AND BYTE INSTRUCTION IS EXECUTED TO VERIFY TH
S13.01 ADDS THE CORRECT SOURCE CONSTANT.

IF FORK A FAILS EXECUTION WILL GO TO EXC.80.
EXC.80 WILL HANG THE PROCESSOR IN THE PAUSE STATE AT MIC
THIS WILL ONLY HAPPEN IF RA CL RADRO1
IS NOT GOING LOW DUE TO RACF A1 RAB01 (AFIR10(1))*U/CLASS

ROM FLOW-22,27,205

1752 TEST 30 ALU CARRY FUNCTIONAL TEST

THIS TEST DOES A COMPLETE CHECK OF THE ALU CARRY FUNCTIO
THE FIRST SECTION ENSURES THAT ALL THE INPUT AND OUTPUT
OK AND THE REST OF THE TEST ENSURES THAT THE CARRY LOGIC
FOLLOWING ARE THE LOGIC EQUATIONS FOR A 74S181 AND 74S18
DICTATED THE PATTERNS USED IN EACH SECTION:

74S181

$$G=A3*B3+A2*B2*(A3+B3)+A1*B1*(A2+B2)*(A3+B3)+A0*B$$

E03

POP 11/70 CPU DIAGNOSTIC PART 1 MACY11 27(732) 03-NOV-76 11:18 PAGE 31
DEKBAB.P11

1174



F03

PDP 11/70 CPU DIAGNOSTIC PART 1

DECDOC 00.0

1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230

$P=(A3+B3)*(A2+B2)*(A1+B1)*(A0+B0)$
COUT=G+P*CIN

745182

CX=G0+P0*CIN
CY=G1+P1*G0+P1*P0*CIN
CZ=G2+P2*G1+P2*P1*G0+P2*P1*P0*CIN

1882 TEST 31 THREE MICROSTATES (DAC*DM2*0/CLASS)

IF FORK A FAILS EXECUTION WILL GO TO RSD.00. THIS WILL C
IF RACE AO RAB01 DOES NOT GO LOW OR DOES NOT GET THRU
TO RACL RAD01. THIS WILL CAUSE A TRAP TO LOCATION 10.

IF BEN15 FAILS EXECUTION WILL GO TO D45.80.
THIS WILL CAUSE A TRAP TO 4 WITH THE ADDRESS OF 1\$ ON TH

IF THE DESTINATION CONSTANT FAILS(ADDS 1 OR 3) IN STATE
AN ODD ADDRESS TRAP WILL OCCUR.
IF THE DST CONST ADDS 0, THE ERROR AT 3\$ WILL REPORT THE
IF THE DESTINATION IS NOT LOADED WITH THE SOURCE, THE
ERROR AFTER 5\$ WILL REPORT THE FAILURE.

ROM FLOW-2,155,312

1943 TEST 32 THREE MICROSTATES (DAC*DM1*0/CLASS)

THIS TEST IS THE SAME AS THE PREVIOUS TEST EXCEPT FOR TH
IF FORK A FAILS, EXECUTION WILL GO TO RSD.00.
THIS WILL CAUSE A TRAP TO LOCATION 10 WITH AN ODD ADDRES
THIS WILL ONLY HAPPEN IF RACE AO RAB00 DOES NOT GO LOW 0
DOES NOT GET TO RACL.
EITHER E44 OR E6 IS BAD.

IF THE INSTRUCTION FAILS TO MOVE R5 TO THE PC INDIRECT,
THE ERROR AFTER 1\$ WILL REPORT THE FAILURE.

ROM FLOW-1,155,312

1976 TEST 33 THREE MICROSTATES (DAC*DM4*0/CLASS)

IF FORK A FAILS EXECUTION WILL GO TO RSD.00.
THIS WILL CAUSE A TRAP TO LOCATION 10. THIS WILL ONLY HA
RACE AO RAB02 IS NOT GOING LOW. EITHER RACE E33 OR
E6(9&8) IS BAD.

IF THE DST CONST FAILS TO SUBTRACT 2, THE ERROR AT EITHE
OR 1\$-2 WILL REPORT THE FAILURE.

IF BEN01 FAILS (CAUSED BY IRCD DM357 STUCK HIGH)
EXECUTION WILL GO TO D10.00 WHICH WILL EXECUTE A MODE 5
INSTEAD OF MODE 4.

IF THE DESTINATION IS NOT LOADED PROPERLY,
THE ERROR AT 3\$ WILL REPORT THE FAILURE.

G03

1231

1232
1233
1234
1235 ROM FLOW-4,122,157
1236
1237 2037 TEST 34 THREE MICROSTATES (DAC*DM1*TST.B*DRO(0))
1238 IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TR
1239 THIS WILL ONLY HAPPEN IF RA SAME?
1240
1241
1242 2055
1243 IF BENIS FAILS EXECUTION WILL GO TO STATE D12.60:
1244
1245 IF B FORK FAILS (AFTER STATE D12.10) CAUSED BY IRCB
1246 K/CLASS STUCK LOW EXECUTION WILL GO TO STATE RSD.00 CAUS
1247 IF IRCB B1 RAB00 IS STUCK HIGH EXECUTION WILL GO FROM
1248 D12.10 TO JSR.40. THIS WILL CAUSE THE PROCESSOR TO HANG.
1249 IF EITHER GRAB 0BD(1) IS STUCK HIGH OR NOT GETTING THRU
1250 EXECUTION WILL GO TO STATE D12.30.
1251 IF GRAB DRMX00 IS STUCK HIGH OR GRAB E50 IS BAD, EXECUTI
1252 WILL GO TO D10.60 WHICH WILL HANG THE PROCESSOR.
1253
1254 ROM FLOW-1,175,33
1255
1256 2093 TEST 35 THREE MICROSTATES (DAC*DM1*BIT.B*DRO(0))
1257
1258 THIS TEST IS THE SAME AS THE PREVIOUS TEST EXCEPT A BIT
1259 IF FORK A FAILS RACE BIN*SMD H FAILED.
1260
1261 IF FORK B FAILS THE INSTRUCTION DECODE ROM WORD IS BAD.
1262
1263 IF THE RESULTANT DATA IS BAD STATE TST.10 FAILED.
1264
1265 ROM FLOW-1,175,33
1266
1267 2116 TEST 36 THREE MICROSTATES (DAC*DM1*CMP.B*DRO(0))
1268
1269 THIS TEST IS THE SAME AS THE PREVIOUS TWO TESTS
1270 EXCEPT A CMP INSTRUCTION IS USED.
1271
1272 ROM FLOW-1,175,33
1273
1274 2135 TEST 37 THREE MICROSTATES (DAC*DM2*TST.B*DRO(0))
1275
1276 IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TR
1277 THIS WILL ONLY HAPPEN IF RACE E45 IS BAD (AFIRO4(1)*R/CL
1278
1279 BENIS & FORK B HAVE ALREADY BEEN TESTED
1280 IF THE AUTO INC FAILS IT WILL BE DUE TO A BAD
1281 FIELD IN ROM STATE D12.10.
1282
1283 ROM FLOW-2,175,33
1284

I03

PDP 11/70 CPU DIAGNOSTIC PART 1 MAC11 27(732) 03-NOV-76 11:18 PAGE 35
DEKBAB.P11

PDP 11/70 CPU DIAGNOSTIC PART 1

DECDOC VER 03.0

1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340

2165 THE LOGICAL SEQUENCE WOULD NEXT TEST THE B.7.B AND CMP.B INSTRUC
THESE WILL NOT BE TESTED WITH INDIVIDUAL TESTS SINCE THEY DO NOT
ANY HARDWARE THAT HAS NOT ALREADY BEEN TESTED.

2172 TEST 40 THREE MICROSTATES (JMP*DM1)
IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TR
THIS WILL ONLY HAPPEN IF RACE AD RAB00 DOES NOT GO LOW
(AFIRO3(1)*[JMP+JSR+SWAB]).

2178 A FORK B FAILURE WOULD BE ONE OF THE FOLLOWING:
IF IRCB (JMP+JSR) IS STUCK LOW EXECUTION WILL GO TO RSD.
A TRAP TO 10.
IF IRCB IR(14:9) 04 IS STUCK LOW EXECUTION WILL
GO TO JSR.00 WHICH WILL EXECUTE A JSR INSTEAD OF A JMP.
IF IRCB B FORK MUX FAILS EXECUTION WILL GO TO
FOP.00.
IF IRCB FJ CLASS IS STUCK HIGH EXECUTION WILL GO TO
STATE D12.00 AND THE JMP WON'T JUMP.
IF THE INSTRUCTION FAILS TO JUMP, STATE JMP.00 IS REPORT
ROM FLOW-1,135,35

2212 TEST 41 THREE MICROSTATES (JMP*DM2)
THIS TEST IS THE SAME AS THE PREVIOUS TEST EXCEPT THE DM
IF FORK A FAILS RAC E45 IS BAD (AFIRO4(1)*[JMP+JSR+SWAB])
ROM FLOW-2,135,35

2229 TEST 42 THREE MICROSTATES (JMP*DM4)
IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TR

2232 THIS WILL ONLY HAPPEN IF RACE E33(AFIRO5(1)*[JMP+JSR+SWA
ALL OTHER LOGIC HAS BEEN TESTED.
ROM FLOW-4,122,35

2246 TEST 43 THREE MICROSTATES (SOB)
IF RACH U/CLASS IS NOT GOING HIGH EXECUTION WILL GO TO R
CAUSING A TRAP TO LOCATION 10 WITH THE ADDRESS OF 55-2
ON THE TOP OF THE STACK.
IF EITHER RACE E10 IS BAD OR RACE BIN DOES NOT GO HIGH
EXECUTION WILL GO TO D67.01. EXECUTION WILL EVENTUALL GO
TO RSD.00 AFTER STATE D10.60 AND THE STACKED PC WILL BE
IF RACF E8 FAILS EXECUTION WILL GO TO ASC.10 WHICH WILL
PERFORM AN ASHC*DMO OPERATION.
IF THE SOB BRANCHES WHEN IT IS NOT SUPPOSE TO EITHER

J03

PDP 11/70 CPU DIAGNOSTIC PART 1 MACY11 27(732) 03-NOV-76 11:18 PAGE 36
DEKBAB.P11

1341

1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397

GRAE SR EQ ONE IS STUCK HIGH OR RACK E63 IS BAD
OR STATE SOB.10 DOES NOT RESTORE THE OLD PC.

IF THE SOB DOES NOT BRANCH WHEN IT IS SUPPOSE TO EITHER
STATE SOB.00 IS BAD OR GRAE SR EQ ONE STUCK LOW OR
RACK E63(C1) IS BAD.

IF THE REGISTER DOES NOT DECREMENT STATE SOB.20 IS BAD.

ROM FLOW-57,242/262.262

2311

2313 TEST 44 FOUR MICROSTATES (DAC*DM12*P/CLASS*DRO(0))

IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TR
THIS WILL ONLY HAPPEN IF RACJ AFIR 14(1) DOES NOT
GET THRU RAC E42.

THE FOLLOWING COULD BE FORK B FAILURES:
IF IRCB 80 RAB00 IS STUCK HIGH OR NOT GETTING THRU
TO RACL RADROO EXECUTION WILL GO TO EXC.90 WHICH WOULD
PUT THE RESULT INTO A REGISTER RATHER THAN MEMORY.
IF IRCB E38(6) IS STUCK HIGH EXECUTION WILL GO TO
RSD.00 CAUSING A TRAP TO 10.
IF THE BIC DOES NOT HAPPEN THEN EXC.00 IS BAD.

ROM FLOW-2,175,31,132

2358 TEST 45 FOUR MICROSTATES (DAC*DM12*TST.B*DRO(1))

AFTER STATE D12.10 IF EITHER GRAB OBD(1) DOES NOT GO HIG
OR DOES NOT GET THRU RACL E71 EXECUTION WILL GO
TO TST.10. IF EITHER GRAB OBD(0) IS STUCK HIGH OR NOT GE
THRU RACK E41, EXECUTION WILL GO TO D10.60. THIS WILL CA
THE PROCESSOR TO HANG UP IN THE PAUSE STATE AT MICRO
ADDRESS 177. IF THE TEST FAILS THEN STATE D12.30 FAILED

ROM FLOW-1,175,137,33

2382 TEST 46 FOUR MICROSTATES (DAC*DM4*TST.B*DRO(0))

IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TR
THIS WILL ONLY OCCUR IF RACE E33(AFIRG5(1)*R/CLASS) IS B

AFTER D10.30 IF IRCB FJ/CLASS DOES NOT GET TO RACL E71
OR IF RACL E71 IS BAD EXECUTION WILL GO TO SVC.50.

IF THE INSTRUCTION DOESN'T WORK THEN D10.60 IS BAD.

ROM FLOW-4,122,177,33

PDP 11/70 CPU DIAGNOSTIC PART 1

DECDOC VER 00.0

1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453

- 2414 THE LOGICAL SEQUENCE HERE WOULD BE TO TEST THE BIT & CMP INSTRU
BUT NO ADDITIONAL LOGIC IS TESTED BY THEM.
- 2420 TEST 47 FOUR MICROSTATES (DAC*DM6*)/CLASS)
FORK A SHOULD NOT FAIL ON THIS TEST.
BEN01 SHOULD NOT FAIL.
BEN15*FEN2 SHOULD NOT FAIL.
IF D67.00 FAILS TO INCREMENT THE PC AN RTI INSTRUCTION
WILL BE EXECUTED.
IF D67.00 FAILS TO CLOCK THE BR THE INSTRUCTION
WILL BE ADDED AS THE INDEX WORD.
IF D67.10 FAILS TO ADD THE INDEX NUMBER THE SOURCE
WILL BE PUT IN THE WRONG LOCATION.
ROM FLOW-6,251,122,157
- 2461 TEST 50 FOUR MICROSTATES (BIN*SM12*DM0*-DF7*SRO(1))
IF FORK A FAILS EXECUTION WILL GO TO STATE D12.00.
THIS WILL HAPPEN IF RACE B:1=7 DOES NOT GO HIGH.
STATE D12.00 WOULD BITB R5 & THE CONTENTS OF LOCATION 20
WHICH IS 137.
THE FOLLOWING COULD BE BEN14*FORK C FAILURES:
IF IRCC CD RAB00 DOES NOT GO HIGH EXECUTION WILL GO TO
D00.90 WHICH WILL TEST THE LOW BYTE INSTEAD OF THE HIGH
IF STATE D00.80 FAILS TO SWAP THE BYTES IT WILL LOOK LIK
A FORK C FAILURE.
ROM FLOW-21,27,204,205
- 2499 TEST 51 FOUR MICROSTATES (BIN*SM12*DM0*DF7*SRO(0))
IF FORK A FAILS EXECUTION WILL EITHER GO TO STATE D12.00
- 2502 STATE D12.00 WOULD ADD R5 TO THE CONTENTS OF THE PC.
STATE EXC.80 WOULD NOT CHANGE THE PC.
IF FORK C FAILS EXECUTION WILL EITHER GO TO JSR.10 OR AS
JSR.10 WILL CAUSE R5 TO BE STACKED, THE PC TO BE PUT
IN R5, AND THE PC REPLACED BY R5 WHICH WILL CAUSE AND RT
THE CONTENTS OF THE LOCATION POINTED TO BY R5 IS 000002.
ASC.80 WILL CAUSE THE ADD TO LOOK LIKE IT FAILED.
IF STATE D07.10 FAILS TO LOAD THE SHFTR THE PC WILL
DOUBLE AND THE PROGRAM WILL BLOW UP.
IF THE SHFTR FAILS TO BE PUT IN THE SR THE PC
WILL NOT CHANGE.

M03

1454

1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505

ROM FLOW-21,27,203,30

2567 TEST 52 FOUR MICROSTATES (BIN*SM12*DM0*DF7*SR0(1))
IF FORK A FAILS EXECUTION WILL GO TO D12.00.
FORK C SHOULD NOT FAIL SINCE THE LOGIC HAS ALREADY BEEN
IF STATE D07.00 FAILS TO SWAP THE BYTES THE CC'S WILL
BE BAD.
IF D07.00 FAILS TO LOAD THE SHIFTER, THE THIRD CMPB WILL
IF D07.00 FAILS TO LOAD THE SR THE SECOND CMPB WILL FAIL

ROM FLOW-21,27,202,30

2604 TEST 53 FOUR MICROSTATES (BIN*SM4*DM0*-DF7*SR0(0))
IF FORK A FAILS EXECUTION WILL
GO TO D45.00 WHICH WILL EXECUTE A SMO*DM4 INSTRUCTION EX
THE DESTINATION REGISTER WILL NOT DECREMENT.

2610 FORK C WILL NOT FAIL SINCE IT HAS ALREADY BEEN TESTED.
IF THE SRC FAILS TO AUTO DECREMENT STATE S45.00 IS BAD.
ROM FLOW-24,23,27,205

2639 TEST 54 FOUR MICROSTATES (RTS)
IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TR
THIS WILL ONLY HAPPEN IF RACF E3 DOES NOT GO HIGH.
IF THE PC OR R5 FAILS THE TEST WILL HALT.
ROM FLOW-40,223,224,342

2661 TEST 55 FOUR MICROSTATES (JMP*DM6)
IF FORK A FAILS EXECUTION WILL GO TO STATE D12.01.

2664 THIS WOULD CAUSE A JMP*DM1 TO EXECUTE.
NEITHER BEND1 NOR BEN15*FEN2 SHOULD FAIL SINCE THEY HAVE
TESTED.
ROM FLOW-6,251,122,35

1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561

2681

2683 TEST 56 FIVE MICROSTATES (DAC*DM12*P/CLASS*DR0(1))

FORK A SHOULDN'T FAIL.

BEN15 SHOULDN'T FAIL SINCE THIS LOGIC HAS BEEN TESTED.
NEITHER SHOULD BEN05*FEN2.

IF FORK B FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TR
THIS FAILURE WOULD BE CAUSED BY A BAD FIELD (PART PCLASS
IN THE IR DECODE ROM.

POM FLOW-1,175,137,31,132

2708 TEST 57 FIVE MICROSTATES (DAC*DM3*0/CLASS)

FORK A SHOULD NOT FAIL SINCE THE LOGIC HAS ALREADY BEEN

IF THE DR DOES NOT AUTO INC THEN STATE D30.10 IS BAD.

IF THE CONDITION CODES ARE BAD THEN EITHER STATE D10.50
DID NOT LOAD THE BR OR THE DOUBLE DEFERED DIDN'T WORK.

ROM FLOW-3,221,233,311,157

2741 TEST 60 FIVE MICROSTATES (DAC*DM4*P/CLASS*DR0(0))

FORK A SHOULD NOT FAIL.

IF FORK B FAILS, AFTER D10.60, EXECUTION WILL GO TO
RSD.00 CAUSING A TRAP TO 10. THIS WILL ONLY HAPPEN IF
THE IR DECODE ROM HAS A BAD FIELD (PART PCLASS).

ROM FLOW-4,122,177,31,132

2765 THE LOGICAL SEQUENCE AT THIS POINT WOULD BE TO EXECUTE A DAC*DM4
BIT.B+CMP.B)*DR0(1) INSTRUCTION FOLLOWED BY A DAC*DM6*[TST.B+BIT
DR0(0) INSTRUCTION TEST BUT NO ADDITIONAL LOGIC IS TESTED.

2774 THE LOGICAL SEQUENCE AT THIS POINT WOULD BE TO EXECUTE A BIN*SM4
FOLLOWED BY A BIN*SM4*DM0*DF7*SR0(0)
FOLLOWED BY A BIN*SM4*DM0*DF7*SR0(1) INSTRUCTION: BUT NO ADDITION

2781 TEST 61 FIVE MICROSTATES (BIN*SM6*DM0*-DF7*SR0(0))

IF FORK A FAILS EXECUTION WILL GO TO STATE D67.00
WHICH WOULD EXECUTE A SMO*DM6 INSTRUCTION.

BEN14*FEN4 SHOULD NOT FAIL SINCE IT HAS ALREADY BEEN TES

ROM FLOW-26,54,141,142,205

C04

POP 11/70 CPU DIAGNOSTIC PART 1 MACY11 27(732) 03-NOV-76 11:18 PAGE 42
DEKBAB.P11

1562

PDP 11/70 CPU DIAGNOSTIC PART 1

DECD0C VER 00.0

1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618

2807 TEST 62 FIVE MICROSTATES (BIN*SM12*DM12*0/CLASS)

IF FORK A FAILS EXECUTION WILL GO TO D12.01. THIS WOULD C
TO BE WRITTEN INTO STMP2.

IF FORK C FAILS EXECUTION WOULD GO TO ONE OF THE
FOLLOWING STATES: D45.80, D30.80, FOP.00, WAT.00, D00.90, A
STATE D45.80 WOULD EXECUTE A SM2*DM4 INSTEAD OF SM2*DM1.
STATE D30.80 WOULD EXECUTE A SM1*DM3.
STATE FOP.00 WOULD CAUSE A TRAP TO LOCATION 10.
THIS WILL ONLY HAPPEN IF EITHER IRCC CO RAB03 IS STUCK
OR IT IS NOT GETTING THRU RACL RAD03 OR
IRCC FORK C MUX INPUT B0 IS HIGH.
THE LA30 PRINTER BUFFER WILL BE SET UP TO GENERATE AN IN
CASE THE TEST FAILS TO THE WAT.00 STATE.
STATE D00.90 WILL EXECUTE A DMO INSTEAD OF A DM1.
STATE ASH.20 WILL CLEAR THE C BIT.

ROM FLOW-22,27,111,155,312

2900 THE LOGICAL FLOW AT THIS POINT WOULD TEST A BIN*SM12*DM12*SRO(0)
*[TST.B+BIT.B+CMP.B] INSTRUCTION BUT NO ADDITIONAL LOGIC WOULD B

2906 TEST 63 FIVE MICROSTATES (BIN*SM12*DM12*SRO(1)*DRO(0)*CMPB)

THE ONLY THING THAT SHOULD FAIL WOULD BE STATE D12.90(11

ROM FLOW-21,27,110,175,33

2925 TEST 64 FIVE MICROSTATES (BIN*SM12*DM4*0/CLASS)

WHICH WILL EXECUTE A SM1*DM2 TYPE INSTRUCTION.

IF THE INSTRUCTION FAILS EITHER D45.80 OR D40.20 FAILED.

ROM FLOW-21,27,115,121,157

2950

2952 TEST 65 SIX MICROSTATES (DAC*DM12*ASRB*DRO(1))

NEITHER FORK A NOR BEN15 NOR BEN05*FEN2 SHOULD FAIL
SINCE THEY HAVE ALREADY BEEN TESTED.

IF FORK B FAILS AFTER D12.30 EXECUTION WILL GO TO
ONE OF THE FOLLOWING: RSD.00, D45.00, EXC.00, S45.00,
CCP.00, MUL.00, SVC.10, MFP.00 OR DEP.00.
RSD.00 WILL CAUSE A TRAP TO LOCATION 10.
THIS WILL HAPPEN IF THE B FORK MUX SELECT IS STUCK LOW.
IF STATE D45.00 IS ENTERED THE PROCESSOR WILL HANG UP
IN A LOOP BETWEEN STATES D45.00 AND D10.30.
IF STATE S45.00 IS ENTERED EXECUTION WILL GO TO STATE

E04

PDP 11/70 CPU DIAGNOSTIC PART 1 MACY11 27(732) 03-NOV-76 11:18 PAGE 44
DEKBAB.P11

1619

1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675

D12.80 AFTER S13.10 WHICH WILL HANG UP THE PROCESSOR.
STATE CCP.00 WOULD SET OR CLEAR THE CONDITION CODES
ACCORDING TO IR(4:0).
STATE MUL.00 WOULD CAUSE THE DESTINATION OPERAND TO
BE MULTIPLIED BY REGISTER 2 AND THE RESULT WOULD BE
STORED IN REGISTER 2 AND 3.
IF SVC.10 IS ENTERED A TRAP TO 4 WILL OCCUR BECAUSE
THE DESTINATION REGISTER IS ODD. THIS WILL ONLY HAPPEN
IF EITHER B FORK MUX INPUT B3 OR IRCB B0 RAB00 IS STUCK
IF MFP.00 IS ENTERED AN MFPI INSTRUCTION WILL BE EXECUTE
THIS WILL PUSH THE ADDRESS OF 1\$ ONTO THE STACK.
DEP.00 WILL CAUSE THE PROCESSOR TO HANG IN MICRO ADDRESS
170 WITH THE RUN LIGHT ON.

ROM FLOW-1,175,137,64,123,132

3028 TEST 66 SIX MICROSTATES (DAC*DM12*RORB*DR0(1))

THIS TEST IS THE SAME AS THE LAST ONE EXCEPT A RORB
IS USED INSTEAD OF AN ASRB.

FORK B WILL ONLY FAIL IF IRCB E36(13) IS STUCK HIGH
WHICH WILL CAUSE EXECUTION TO GO TO EXC.00.

ROM FLOW-2,175,137,64,123,132

3053 THE LOGICAL FLOW AT THIS POINT WOULD TEST A DAC*DM3*[TST.B+BIT.B
FOLLOWED BY A DAC*DM4*P/CLASS*DR0(0) INSTRUCTION BUT NO ADDITION
IS TESTED

3061 TEST 67 SIX MICROSTATES (DAC*DM6*XOR*DR0(0))

IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TR
THIS SHOULD ONLY HAPPEN IF RACE E35(1) IS BAD.

IF THE INSTRUCTION DOESN'T WORK IT WILL HALT IN THIS TES

ROM FLOW-6,251,122,177,31,132

3086 THE LOGICAL SEQUENCE WOULD TEST A DAC*DM6*[TST.B+BIT.B+CMP.B]*DR
THE LOGIC HAS BEEN TESTED.

3093 TEST 70 SIX MICROSTATES (NEG.B*DM12*DR0(0))

NEITHER FORK A NOR BGN15 SHOULD FAIL.

3096

IF FORK B FAILS EXECUTION WILL GO TO RSD.00 CAUSING
A TRAP TO LOCATION 10 OR EXC.00.
RSD.00 SHOULD ONLY OCCUR IF THE B FORK MUX
STROBE IS BEING HELD LOW(CHIP FAILURE).

ROM FLOW-1,175,67,271,163,132

PDP 11/70 CPU DIAGNOSTIC PART 1

DECDOC VER 00.0

1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728

- 3135 THE LOGICAL SEQUENCE WOULD NEXT EXECUTE A JMP*DM3 BUT NO ADDITIONAL LOGIC IS TESTED.
- 3141 TEST 71 SIX MICROSTATES (BIN*SM3*DM0*-DF7*SRO(0))
FORK A SHOULD NOT FAIL.
IF BEN14*FEN4 FAILS EXECUTION WILL GO TO D00.90. THIS WILL CAUSE A SM2 TO BE EXECUTED. THIS SHOULD ONLY HAPPEN IF IRCC SM357 IS STUCK LOW OR NOT GETTING THRU RA
ROM FLOW-22,27,317,143,146,205
- 3170 THE LOGICAL SEQUENCE WOULD NEXT TEST A BIN*SM6*DM0*DF7*SRO(1), T DM12*SRO(0)*DRO(1)*[TST.B+BIT.B+CMP.B] THEN A BIN*SM12*DM12*SRO(DRO(0)*P/CLASS THEN A BIN*SM12*DM12*SRO(1)*DRO(1)*[TST.B+BIT.B+C THEN A BIN*SM12*DM12*SRO(1)*DRO(0)*P/CLASS AND THEN A BIN*SM12*D SRO(0)*DRO(0)*[TST.B+BIT.B+CMP.B] INSTRUCTION, BUT NO ADDITIONAL IS TESTED.
- 3180 TEST 72 SIX MICROSTATES (BIN*SM12*DM4*SRO(1)*DRO(0)*CMPB)
A FAILURE WILL ONLY OCCUR IF STATE D45.90 FAILS.
IF THE DST REG. DOES NOT DECREMENT STATE D45.90 IS BAD.
IF THE CONDITION CODES ARE BAD THEN STATE D40.30 PROBABLY DID NOT SWAP THE BYTES OF THE SRC OPERAND.
ROM FLOW-1,27,114,131,177,33
- 3212 THE LOGICAL FLOW WOULD NEXT TEST A BIN*SM4*DM12*SRO(0)*DRO(0)*0/ THEN A BIN*SM4*DM12*SRO(0)*DRO(0)*[TST.B+BIT.B+CMP.B] THEN A BIN DM12*SRO(1)*DRO(0)*[TST.B+BIT.B+CMP.B] THEN A BIN*SM4*DM4*SRO(0) 0/CLASS INSTRUCTION, BUT NO ADDITIONAL LOGIC IS TESTED.
- 3220 TEST 73 SIX MICROSTATES (DAC*DMS*DRO(0)*0/CLASS)
FORK A SHOULD NOT FAIL.
IF IRCD DM357 IS STUCK LOW OR NOT GETTING THRU TO RACK E51 A DM4 WILL BE EXECUTED.
IF STATE D10.00 OR D10.10 FAIL TO FETCH THE DEFERED ADDR THE SOURCE WILL BE STORED IN THE DESTINATION.
ROM FLOW-5,162,231,233,311,157
- 3252 THE LOGICAL SEQUENCE WOULD NEXT TEST A DAC*DM3*DRO(0)*P/CLASS TH DAC*DM3*DRO(1)*[TST.B+BIT.B+CMP.B] THEN A DAC*DM4*DRO(1)*[ASRB+ RORB] THEN A DAC*DMS*DRO(0)*[TST.B+BIT.B+CMP.B] THEN A DAC*DM6* DRO(1)*P/CLASS INSTRUCTION, BUT NO ADDITIONAL LOGIC IS TESTED.

1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780

3259

3261 TEST 74 SEVEN MICROSTATES (DAC*DM7*0/CLASS)

FORK A SHOULD NOT FAIL.

IF IRCD DM357 DOES NOT GO HIGH A DM6 WILL BE EXECUTED.

ALL OTHER LOGIC HAS BEEN TESTED.

ROM FLOW-7,251,162,231,233,311,157

3292 THE LOGICAL SEQUENCE WOULD NEXT TEST A NEG.B*DM12*DR0(1) THEN A THEN A JMP*DMS INSTRUCTION, BUT NO ADDITIONAL LOGIC IS TESTED.

3298 TEST 75 SEVEN MICROSTATES (JSR*DM12)

IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TR THIS WILL ONLY OCCUR IF RACE JMP+JSR+SWAB DOES NOT GO HI

IF EITHER IRCB IR(14:9)04 DOES NOT GO LOW OR E63 IS BAD EXECUTION WILL GO FROM D12.10 TO EXC.00.
IF IRCB E63 IS BAD (PIN 10 OR 4&5 FLOATING) EXECUTION WI GO TO RSD.00 CAUSING A TRAP TO LOCATION 10. THIS FAILURE WOULD INCREMENT THE DST REG. BEFORE THE TRAP.

IF THE INSTRUCTION FAILS THEN ONE OF THE JSR STATES FAIL

ROM FLOW-2,135,34,201,274,275,32

3342 THE LOGICAL SEQUENCE WOULD NEXT EXECUTE A BIN*SM3*DM0*-DF7*SRO(1 A BIN*SM3*DM0*DF7*SRO(0) THEN A BIN*SM3*DM0*DF7*SRO(1) INSTRUCTI BUT NO ADDITIONAL LOGIC IS TESTED.

3349 TEST 76 SEVEN MICROSTATES (BIN*SMS*DM0*-DF7*SRO(0))

FORK A SHOULD NOT FAIL.

IF BEN14*FEN4 FAILS EXECUTION WILL GO TO D00.90 CAUSING A SM4 INSTRUCTION TO BE EXECUTED. THIS WILL ONLY OCCUR IF EITHER IRCC SRCMS DOES NOT GO LOW OR IF IRCC E2

ROM FLOW-24,23,27,317,143,146,205

3378 THE LOGICAL SEQUENCE WOULD NEXT EXECUTE A BIN*SM12*DM12*SRO(0)*D P/CLASS THEN A BIN*SM12*DM12*SRO(1)*DR0(1)P/CLASS INSTRUCTION, B NO ADDITIONAL LOGIC IS TESTED.

1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836

3385 TEST 77 SEVEN MICROSTATES (BIN*+SM12*DM3*0/CLASS)
IF IRCC C FORK MUX INPUT B2 IS NOT GOING LOW OR IRCC E40 IS BAD A DM2 WILL BE EXECUTED.
THE ONLY OTHER POSSIBLE FAILURE IS STATE D30.80.
ROM FLOW-21,27,113,221,233,311,157

3414 THE LOGICAL SEQUENCE WOULD NEXT TEST A BIN*SM12*DM4*SRO(0)*DRO(0) P/CLASS FOLLOWED BY A BIN*SM12*DM4*SRO(0)*DRO(1)*[TST.B+BIT.B+CM FOLLOWED BY A BIN*SM12*DM4*SRO(1)*DRO(0)*P/CLASS FOLLOWED BY A BIN*SM12*DM4*SRO(1)*DRO(1)*[TST.B+BIT.B+CM] INSTRUCTION, BUT ADDITIONAL LOGIC IS TESTED.

3423 TEST 100 SEVEN MICROSTATES (BIN*SM12*DM6*0/CLASS)
IF FORK C FAILS EXECUTION WILL GO TO D45.90 AND A DM4 WILL BE EXECUTED. THIS WILL ONLY HAPPEN IF IRCC E39 IS NOT GOING LOW. THIS WILL CAUSE AN RTI SINCE THE LOCAT FOLLOWING THE INSTRUCTION CONTAINS 000002.
THE ONLY OTHER FAILURE WOULD BE CAUSED BY STATE D67.80 B
ROM FLOW-21,27,117,6,251,122,157

3454 THE LOGICAL SEQUENCE WOULD NEXT TEST THE INSTRUCTIONS BETWEEN BIN*SM4*DM12*SRO(0)*DRO(1)*[TST.B+BIT.B+CM] AND BIN*SM5*DM0*D BUT NOT ADDITIONAL LOGIC IS TESTED.

3460

3462 TEST 101 EIGHT MICROSTATES (BIN*SM7*DM0*-DF7*SRO(0))
FORK A SHOULD NOT FAIL.
IF FEN4*BEN14 FAILS EXECUTION WILL GO TO D00.90 CAUSING A SM6 TO BE EXECUTED. THIS WILL ONLY HAPPEN IF EITHER IRCC SRCM7 DOES NOT GO LOW OR IF IRCC E28(1) IS BAD.
ROM FLOW-26,54,141,142,317,143,146,205

3492 THE LOGICAL SEQUENCE WOULD NEXT TEST A BIN*SM12*DM3*SRO(0)*DRO(0) BIT.B+CM] BUT NO ADDITIONAL LOGIC IS TESTED.

3498 TEST 102 EIGHT MICROSTATES (BIN*SM12*DM3*SRO(1)*DRO(0)*CM
THE ONLY POSSIBLE FAILURE WOULD BE IN STATE D30.90 SINCE ALL THE OTHER LOGIC HAS BEEN TESTED.
ROM FLOW-21,27,112,221,233,311,177,33

1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888

3520 THE LOGICAL SEQUENCE WOULD NEXT TEST INSTRUCTIONS BIN*SM12*DM4*S
P/CLASS THRU BIN*SM12*DM6*SRO(0)*DRO(0)*[TST.B+BIT.B+CMF.B] BUT
ADDITIONAL LOGIC IS TESTED.

3527 TEST 103 EIGHT MICROSTATES (BIN*SM12*DM6*SRO(1)*DRO(0)*CM

3528 THE ONLY POSSIBLE FAILURE IN THIS TEST IS STATE D67.90.
ROM FLOW-21,27,116,6,251,122,177,33

3543 *****

3545 TEST 104 NINE MICROSTATES (BIN*SM12*DM5*SRO(0)*DRO(0)*CM
THE ONLY POSSIBLE FAILURE IN THIS TEST IS STATE D50.20.
ROM FLOW-21,27,115,161,231,233,311,177,33

3564 TEST 105 EIGHT MICROSTATES (BIN*SM12*DM5*SRO(1)*DRO(0)*CM
THE ONLY POSSIBLE FAILURE IN THIS TEST IS STATE D50.30.

3580 TEST 106 WRITE/READ PSW

3582 THIS TEST VERIFIES THAT THE PSW CAN BE READ THRU THE DAT
IF THE TEST FAILS ONE OF MANY THINGS COULD BE BAD WHICH
DETERMINED IN THIS DIAGNOSTIC.
THIS TEST REQUIRES THAT SCCE PS ADRS GETS TO TMC,
THAT THE TMC DMUX SELECT LINES GET TO PDR, THAT THE PSW
BITS GET TO THE DMUX, THAT SCCE INTERNAL ADDRESS GETS TO
AND SCCA VAOO GETS TO UBC.

3615 TEST 107 RTI
IF FORK A FAILS EXECUTION WILL GO TO ONE OF THREE STATES
RSD.00 WILL CAUSE A TRAP TO LOCATION 4. THIS WOULD HAPPE
IF RACF (HALT:OP CD 7) DOES NOT GO HIGH.
STATE D12.01 WOULD CAUSE AN ODD ADDRESS TRAP SINCE RD
WILL CONTAIN A 1. THIS WILL HAPPEN IF RACE E7 IS BAD.
HLT.00 WILL CAUSE THE PROCESSOR TO HALT ON THE INSTRUCTI
UNDER TEST AND WILL OCCUR IF RACF E17 IS BAD.
IF THE INSTRUCTION DOESN'T WORK THEN ONE OF THE RTI MACH
STATES IS BAD.
ROM FLOW-12,156,212,213,214,215,172

PDP 11/70 CPU DIAGNOSTIC PART 1

DECDOC VER 00.0

1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929

3652 THE RTT WILL NOT BE TESTED HERE SINCE THE ONLY POSSIBLE FORK A FAILURE WOULD CAUSE AN RTI TO BE EXECUTED. THE T BIT FUNCTIONS O THE RTI & RTT ARE TESTED IN PART 2.

3659 TEST 110 EMT AND TRAP
FORK A SHOULD NOT FAIL.
THE INSTRUCTIONS ARE EXECUTED AND THE STACK IS CHECKED T VERIFY THAT EVERYTHING WORKED OK.

ROM FLOW-0,345,354,SVC.00-SVC.90

3741 TEST 111 IOT
FORK A SHOULD NOT FAIL.

3744
IF THE TRAP VECTOR LOGIC FAILS THE TRAP VECTOR COULD COM OUT TO BE 0, 4, OR 24.

THE ONLY OTHER POSSIBLE FAILURE WOULD BE STATE TRP.01. IF THIS STATE FAILS TO LOAD THE DR THE TRAP VECTOR WILL BE WHATEVER IS IN R4. IF IT FAILS TO LOAD THE BR THE OL PS WILL FAIL TO BE STACKED.

ROM FLOW-14,354,(SVC.00-SVC.90) 355,65,357,360,367,37,25

3815 *****
END OF PASS ROUTINE

3817 INCREMENT THE PASS NUMBER (\$PASS)
INDICATE END-OF-PROGRAM AFTER 144 PASSES THRU THE PROGRAM
TYPE "END PASS"
IF THERES A MONITOR GO TO IT
IF THERE ISN'T JUMP TO TST1
IF IT IS DESIRED TO HAVE A BELL INDICATE THE "END OF PASS" LOCAT \$ENDMG CAN BE CHANGED TO 7.

1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979

3851

TYPE ROUTINE

3853 ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 B
THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE
NOTE1: \$NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CH
NOTE2: \$FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED
NOTE3: \$FILLC CONTAINS THE CHARACTER TO FILL AFTER.

CALL:
1) USING A TRAP INSTRUCTION
TYPE ,MESADR ;;MESADR IS FIRST ADDRESS OF AN
OR
TYPE
MESADR
2) USING A JSR INSTRUCTION
MOV PS,-(SP) ;;PUSH PROCESSOR STATUS WORD ON
JSR PC,\$TYPE ;;CALL TYPE ROUTINE
MESADDR ;;FIRST ADDRESS OF MESSAGE

3924

BINARY TO OCTAL (ASCII) AND TYPE

3926 THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIG
OCTAL (ASCII) NUMBER AND TYPE IT.
\$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS
CALL:

MOV NUM,-(SP) ;;NUMBER TO BE TYPED
TYPOS ;;CALL FOR TYPEOUT
.BYTE N ;;N=1 TO 6 FOR NUMBER OF DIGITS
.BYTE M ;;M=1 OR 0
;;1=TYPE LEADING ZEROS
;;0=SUPPRESS LEADING ZER

\$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE
\$TYPOS OR \$TYPOC

CALL:
MOV NUM,-(SP) ;;NUMBER TO BE TYPED
TYPON ;;CALL FOR TYPEOUT

\$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER

CALL:
MOV NUM,-(SP) ;;NUMBER TO BE TYPED
TYPOC ;;CALL FOR TYPEOUT

1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009

4002

CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

4004

THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIG SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE T BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS REPLACED WITH SPACES.
CALL:

```
MOV    NUM, -(SP)    ;;PUT THE BINARY NUMBER ON THE S
TYPDS                      ;;GO TO THE ROUTINE
```

4070

TRAP DECODER

4072

THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTIO AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDR OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL GO TO THAT ROUTINE.

4085

TRAP TABLE

4087

THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLE BY THE "TRAP" INSTRUCTION.

N04

!

2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036

160000

.TITLE PDP 11/70 CPU DIAGNOSTIC PART 1
;*COPYRIGHT (C) JULY 21, 1975
;*DIGITAL EQUIPMENT CORP.
;*MAYNARD, MASS. 01754
;*PROGRAM BY DONALD W. MONROE
;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
;*PACKAGE (MAINDEC-11-DZGAC-A5).
\$SWR=160000 ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYP0UT

2037
 2038
 2039
 2040
 2041
 2042
 2043
 2044
 2045
 2046
 2047
 2048
 2049
 2050
 2051
 2052
 2053
 2054
 2055
 2056
 2057
 2058
 2059
 2060
 2061
 2062
 2063
 2064
 2065
 2066
 2067
 2068
 2069
 2070
 2071
 2072
 2073
 2074
 2075
 2076
 2077
 2078
 2079
 2080
 2081
 2082
 2083
 2084
 2085
 2086
 2087
 2088
 2089
 2090
 2091
 2092

001100
 001100
 000700
 000600
 !77776
 177774
 !77772
 177570
 177570
 000011
 000012
 000015
 000200
 000000
 000001
 000002
 000003
 000004
 000005
 000006
 000007
 .EQUIV R0,R10
 .EQUIV R1,R11
 .EQUIV R2,R12
 .EQUIV R3,R13
 .EQUIV R4,R14
 .EQUIV R5,R15
 .EQUIV R6,SP
 .EQUIV SP,KSP
 .EQUIV SP,SSP
 .EQUIV SP,USP
 .EQUIV R7,PC
 000000
 000040
 000100
 000140
 000200
 000240
 000300
 000340
 !00000

.SBTTL BASIC DEFINITIONS

```

; *INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
STACK= 1100          ;; FIRST ADDRESS OF THE STACK
KERSTK= STACK       ;; KERNEL STACK
SUPSTK= STACK-200   ;; SUPERVISOR STACK
USESTK= STACK-300   ;; USER STACK
.EQUIV ENT,ERROR    ;; BASIC DEFINITION OF ERROR CALL
.EQUIV IOT,SCOPE    ;; BASIC DEFINITION OF SCOPE CALL
PS= 177776          ;; PROCESSOR STATUS WORD
.EQUIV PS,PSW
STKLMT= 177774      ;; STACK LIMIT REGISTER
PIRQ= 177772        ;; PROGRAM INTERRUPT REQUEST REGISTER
SWR= 177570         ;; SWITCH REGISTER
DISPLAY=SWR
  
```

; *MISCELLANEOUS DEFINITIONS

```

HT= 11              ;; CODE FOR HORIZONTAL TAB
LF= 12              ;; CODE LINE FEED
CR= 15              ;; CODE CARRIAGE RETURN
CRLF= 200           ;; CODE FOR CARRIAGE RETURN-LINE FEED
  
```

; *GENERAL PURPOSE REGISTER DEFINITIONS

```

R0= %0              ;; GENERAL REGISTER
R1= %1              ;; GENERAL REGISTER
R2= %2              ;; GENERAL REGISTER
R3= %3              ;; GENERAL REGISTER
R4= %4              ;; GENERAL REGISTER
R5= %5              ;; GENERAL REGISTER
R6= %6              ;; GENERAL REGISTER
R7= %7              ;; GENERAL REGISTER
.EQUIV R0,R10      ;; GENERAL REGISTER
.EQUIV R1,R11      ;; GENERAL REGISTER
.EQUIV R2,R12      ;; GENERAL REGISTER
.EQUIV R3,R13      ;; GENERAL REGISTER
.EQUIV R4,R14      ;; GENERAL REGISTER
.EQUIV R5,R15      ;; GENERAL REGISTER
.EQUIV R6,SP       ;; STACK POINTER
.EQUIV SP,KSP      ;; KERNEL STACK POINTER
.EQUIV SP,SSP      ;; SUPERVISOR STACK POINTER
.EQUIV SP,USP      ;; USER STACK POINTER
.EQUIV R7,PC       ;; PROGRAM COUNTER
  
```

; *PRIORITY LEVEL DEFINITIONS

```

PR0= 0              ;; PRIORITY LEVEL 0
PR1= 40             ;; PRIORITY LEVEL 1
PR2= 100            ;; PRIORITY LEVEL 2
PR3= 140            ;; PRIORITY LEVEL 3
PR4= 200            ;; PRIORITY LEVEL 4
PR5= 240            ;; PRIORITY LEVEL 5
PR6= 300            ;; PRIORITY LEVEL 6
PR7= 340            ;; PRIORITY LEVEL 7
  
```

; *"SWITCH REGISTER" SWITCH DEFINITIONS

```

SW15= 100000
  
```

2093 040000
2094 020000
2095 010000
2096 004000
2097 002000
2098 001000
2099 000400
2100 000200
2101 000100
2102 000040
2103 000020
2104 000010
2105 000004
2106 000002
2107 000001

SW14= 40000
SW13= 20000
SW12= 10000
SW11= 4000
SW10= 2000
SW09= 1000
SW08= 400
SW07= 200
SW06= 100
SW05= 40
SW04= 20
SW03= 10
SW02= 4
SW01= 2
SW00= 1

.EQUIV SW09,SW9
.EQUIV SW08,SW8
.EQUIV SW07,SW7
.EQUIV SW06,SW6
.EQUIV SW05,SW5
.EQUIV SW04,SW4
.EQUIV SW03,SW3
.EQUIV SW02,SW2
.EQUIV SW01,SW1
.EQUIV SW00,SW0

.*DATA BIT DEFINITIONS (BIT00 TO BIT15)

2119 100000
2120 040000
2121 020000
2122 010000
2123 004000
2124 002000
2125 001000
2126 000400
2127 000200
2128 000100
2129 000040
2130 000020
2131 000010
2132 000004
2133 000002
2134 000001

BIT15= 100000
BIT14= 40000
BIT13= 20000
BIT12= 10000
BIT11= 4000
BIT10= 2000
BIT09= 1000
BIT08= 400
BIT07= 200
BIT06= 100
BIT05= 40
BIT04= 20
BIT03= 10
BIT02= 4
BIT01= 2
BIT00= 1

.EQUIV BIT09,BIT9
.EQUIV BIT08,BIT8
.EQUIV BIT07,BIT7
.EQUIV BIT06,BIT6
.EQUIV BIT05,BIT5
.EQUIV BIT04,BIT4
.EQUIV BIT03,BIT3
.EQUIV BIT02,BIT2
.EQUIV BIT01,BIT1
.EQUIV BIT00,BIT0

2135 000001
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148 000004

.*BASIC "CPU" TRAP VECTOR ADDRESSES
ERRVEC= 4 ; ; TIME OUT AND OTHER ERRORS

2149	000010	RESVEC= 10	:: RESERVED AND ILLEGAL INSTRUCTIONS
2150	000014	TBITVEC=14	:: "1" BIT
2151	000014	TRTVEC= 14	:: TRACE TRAP
2152	000014	BPTVEC= 14	:: BREAKPOINT TRAP (BPT)
2153	000020	IOTVEC= 20	:: INPUT/OUTPUT TRAP (IOT) **SCOPE**
2154	000024	PWRVEC= 24	:: POWER FAIL
2155	000030	EMTVEC= 30	:: EMULATOR TRAP (EMT) **ERROR**
2156	000034	TRAPVEC=34	:: "TRAP" TRAP
2157	000060	TKVEC= 60	:: TTY KEYBOARD VECTOR
2158	000064	TPVEC= 64	:: TTY PRINTER VECTOR
2159	000114	CACHVEC=114	:: CACHE ERROR INTERRUPT VECTOR
2160	000240	PIRQVEC=240	:: PROGRAM INTERRUPT REQUEST VECTOR
2161	000250	MMVEC= 250	:: MEMORY MANAGEMENT VECTOR

.SBTTL CACHE REGISTER DEFINITIONS

2166	177740	LOADRS = 177740	:: LOWER 16 BITS OF ADDRESS THAT CAUSED ERROR
2167	177742	HIADRS = 177742	:: UPPER SIX BITS OF ADDRESS THAT CAUSED ERROR
2168	177744	MEMERR = 177744	:: CACHE ERROR REGISTER
2169	177746	CONTRL = 177746	:: MEMORY CONTROL REGISTER
2170	177750	MAINT = 177750	:: MEMORY MAINTENANCE REGISTER
2171	177752	HITMIS = 177752	:: HIT MISS REGISTER "1" IMPLIES HIT IN CACHE

.SBTTL CPU REGISTER DEFINITIONS

2177	177760	SIZELO = 177760	:: MEMORY SIZE REGISTER NUMBER TO PUT INTO A PAR
2178			:: TO GET TO THE LAST 32 WORDS OF MEMORY
2179	177762	SIZEHI = 177762	:: HIGH SIZE REGISTER, RESERVED FOR FUTURE USE
2180			:: CURRENTLY ALL ZERO
2181	177764	SYSTID = 177764	:: SYSTEM ID REGISTER
2182	177766	CPUERR = 177766	:: CPU ERROR REGISTER HOLDS CONDITION THAT CAUSED
2183			:: THE TRAP TO ERRVEC (000004)

.SBTTL MEMORY MANAGEMENT DEFINITIONS

;*MEMORY MANAGEMENT STATUS REGISTER ADDRESSES

2193	177572	MMR0= 177572
2194	177574	MMR1= 177574
2195	177576	MMR2= 177576
2196	172516	MMR3= 172516
2197		.EQUIV MMR0,SR0
2198		.EQUIV MMR1,SR1
2199		.EQUIV MMR2,SR2
2200		.EQUIV MMR3,SR3

;*USER "I" PAGE DESCRIPTOR REGISTERS

2204	177600	UIPDRO= 177600
------	--------	----------------

1.0

2205	177602	UIPDR1= 177602
2206	177604	UIPDR2= 177604
2207	177606	UIPDR3= 177606
2208	177610	UIPDR4= 177610
2209	177612	UIPDR5= 177612
2210	177614	UIPDR6= 177614
2211	177616	UIPDR7= 177616
2212		
2213		;*USER "D" PAGE DESCRIPTOR REGISTERS
2214		
2215	177620	UDPDR0= 177620
2216	177622	UDPDR1= 177622
2217	177624	UDPDR2= 177624
2218	177626	UDPDR3= 177626
2219	177630	UDPDR4= 177630
2220	177632	UDPDR5= 177632
2221	177634	UDPDR6= 177634
2222	177636	UDPDR7= 177636
2223		
2224		;*USER "I" PAGE ADDRESS REGISTERS
2225		
2226	177640	UIPAR0= 177640
2227	177642	UIPAR1= 177642
2228	177644	UIPAR2= 177644
2229	177646	UIPAR3= 177646
2230	177650	UIPAR4= 177650
2231	177652	UIPAR5= 177652
2232	177654	UIPAR6= 177654
2233	177656	UIPAR7= 177656
2234		
2235		;*USER "D" PAGE ADDRESS REGISTERS
2236		
2237	177660	UDPAR0= 177660
2238	177662	UDPAR1= 177662
2239	177664	UDPAR2= 177664
2240	177666	UDPAR3= 177666
2241	177670	UDPAR4= 177670
2242	177672	UDPAR5= 177672
2243	177674	UDPAR6= 177674
2244	177676	UDPAR7= 177676
2245		
2246		;*SUPERVISOR "I" PAGE DESCRIPTOR REGISTERS
2247		
2248	172200	SIPDR0= 172200
2249	172202	SIPDR1= 172202
2250	172204	SIPDR2= 172204
2251	172206	SIPDR3= 172206
2252	172210	SIPDR4= 172210
2253	172212	SIPDR5= 172212
2254	172214	SIPDR6= 172214
2255	172216	SIPDR7= 172216
2256		
2257		;*SUPERVISOR "D" PAGE DESCRIPTOR REGISTERS
2258		
2259	172220	SDPDR0= 172220
2260	172222	SDPDR1= 172222

2261	172224	SDPDR2= 172224
2262	172226	SDPDR3= 172226
2263	172230	SDPDR4= 172230
2264	172232	SDPDR5= 172232
2265	172234	SDPDR6= 172234
2266	172236	SDPDR7= 172236
2267		
2268		;*SUPERVISOR "I" PAGE ADDRESS REGISTERS
2269		
2270	172240	SIPAR0= 172240
2271	172242	SIPAR1= 172242
2272	172244	SIPAR2= 172244
2273	172246	SIPAR3= 172246
2274	172250	SIPAR4= 172250
2275	172252	SIPAR5= 172252
2276	172254	SIPAR6= 172254
2277	172256	SIPAR7= 172256
2278		
2279		;*SUPERVISOR "D" PAGE ADDRESS REGISTERS
2280		
2281	172260	SDPAR0= 172260
2282	172262	SDPAR1= 172262
2283	172264	SDPAR2= 172264
2284	172266	SDPAR3= 172266
2285	172270	SDPAR4= 172270
2286	172272	SDPAR5= 172272
2287	172274	SDPAR6= 172274
2288	172276	SDPAR7= 172276
2289		
2290		;*KERNEL "I" PAGE DESCRIPTOR REGISTERS
2291		
2292	172300	KIPDR0= 172300
2293	172302	KIPDR1= 172302
2294	172304	KIPDR2= 172304
2295	172306	KIPDR3= 172306
2296	172310	KIPDR4= 172310
2297	172312	KIPDR5= 172312
2298	172314	KIPDR6= 172314
2299	172316	KIPDR7= 172316
2300		
2301		;*KERNEL "D" PAGE DESCRIPTOR REGISTERS
2302		
2303	172320	KDPDR0= 172320
2304	172322	KDPDR1= 172322
2305	172324	KDPDR2= 172324
2306	172326	KDPDR3= 172326
2307	172330	KDPDR4= 172330
2308	172332	KDPDR5= 172332
2309	172334	KDPDR6= 172334
2310	172336	KDPDR7= 172336
2311		
2312		;*KERNEL "I" PAGE ADDRESS REGISTERS
2313		
2314	172340	KIPAR0= 172340
2315	172342	KIPAR1= 172342
2316	172344	KIPAR2= 172344

2317	172346	KIPAR3= 172346
2318	172350	KIPAR4= 172350
2319	172352	KIPAR5= 172352
2320	172354	KIPAR6= 172354
2321	172356	KIPAR7= 172356

;*KERNEL "D" PAGE ADDRESS REGISTERS

2322		
2323		
2324		
2325	172360	KDPAR0= 172360
2326	172362	KDPAR1= 172362
2327	172364	KDPAR2= 172364
2328	172366	KDPAR3= 172366
2329	172370	KDPAR4= 172370
2330	172372	KDPAR5= 172372
2331	172374	KDPAR6= 172374
2332	172376	KDPAR7= 172376

.SBTTL UNIBUS MAP REGISTER DEFINITIONS

;*THE LOWER 16 BITS OF THE MAP REGISTERS ARE LABELED 'MAPLXX'
;*THE UPPER 6 BITS OF THE MAP REGISTERS ARE LABELED 'MAPHXX'

2333		
2334		
2335		
2336		
2337		
2338		
2339		
2340		
2341		
2342		
2343		
2344	170200	MAPL00 = 170200
2345	170202	MAPH00 = 170202
2346	170204	MAPL01 = 170204
2347	170206	MAPH01 = 170206
2348	170210	MAPL02 = 170210
2349	170212	MAPH02 = 170212
2350	170214	MAPL03 = 170214
2351	170216	MAPH03 = 170216
2352	170220	MAPL04 = 170220
2353	170222	MAPH04 = 170222
2354	170224	MAPL05 = 170224
2355	170226	MAPH05 = 170226
2356	170230	MAPL06 = 170230
2357	170232	MAPH06 = 170232
2358	170234	MAPL07 = 170234
2359	170236	MAPH07 = 170236
2360	170240	MAPL10 = 170240
2361	170242	MAPH10 = 170242
2362	170244	MAPL11 = 170244
2363	170246	MAPH11 = 170246
2364	170250	MAPL12 = 170250
2365	170252	MAPH12 = 170252
2366	170254	MAPL13 = 170254
2367	170256	MAPH13 = 170256
2368	170260	MAPL14 = 170260
2369	170262	MAPH14 = 170262
2370	170264	MAPL15 = 170264
2371	170266	MAPH15 = 170266
2372	170270	MAPL16 = 170270

2373	170272	MAPH16 =	170272
2374	170274	MAPL17 =	170274
2375	170276	MAPH17 =	170276
2376	170300	MAPL20 =	170300
2377	170302	MAPH20 =	170302
2378	170304	MAPL21 =	170304
2379	170306	MAPH21 =	170306
2380	170310	MAPL22 =	170310
2381	170312	MAPH22 =	170312
2382	170314	MAPL23 =	170314
2383	170316	MAPH23 =	170316
2384	170320	MAPL24 =	170320
2385	170320	MAPH24 =	170320
2386	170324	MAPL25 =	170324
2387	170326	MAPH25 =	170326
2388	170330	MAPL26 =	170330
2389	170332	MAPH26 =	170332
2390	170334	MAPL27 =	170334
2391	170336	MAPH27 =	170336
2392	170340	MAPL30 =	170340
2393	170342	MAPH30 =	170342
2394	170344	MAPL31 =	170344
2395	170346	MAPH31 =	170346
2396	170350	MAPL32 =	170350
2397	170352	MAPH32 =	170352
2398	170354	MAPL33 =	170354
2399	170356	MAPH33 =	170356
2400	170360	MAPL34 =	170360
2401	170362	MAPH34 =	170362
2402	170364	MAPL35 =	170364
2403	170366	MAPH35 =	170366
2404	170370	MAPL36 =	170370
2405	170372	MAPH36 =	170372
2406	170374	MAPL37 =	170374
2407	170376	MAPH37 =	170376
2408		.EQUIV	MAPL00, MAPL0
2409		.EQUIV	MAPH00, MAPH0
2410		.EQUIV	MAPL01, MAPL1
2411		.EQUIV	MAPH01, MAPH1
2412		.EQUIV	MAPL02, MAPL2
2413		.EQUIV	MAPH02, MAPH2
2414		.EQUIV	MAPL03, MAPL3
2415		.EQUIV	MAPH03, MAPH3
2416		.EQUIV	MAPL04, MAPL4
2417		.EQUIV	MAPH04, MAPH4
2418		.EQUIV	MAPL05, MAPL5
2419		.EQUIV	MAPH05, MAPH5
2420		.EQUIV	MAPL06, MAPL6
2421		.EQUIV	MAPH06, MAPH6
2422		.EQUIV	MAPL07, MAPL7
2423		.EQUIV	MAPH07, MAPH7
2424			
2425			
2426			
2427			
2428			
2429			
2430			
2431			
2432			
2433			
2434			
2435			
2436			
2437			
2438			
2439			
2440			
2441			
2442			
2443			
2444			
2445			
2446			
2447			
2448			
2449			
2450			
2451			
2452			
2453			
2454			
2455			
2456			
2457			
2458			
2459			
2460			
2461			
2462			
2463			
2464			
2465			
2466			
2467			
2468			
2469			
2470			
2471			
2472			
2473			
2474			
2475			
2476			
2477			
2478			
2479			
2480			
2481			
2482			
2483			
2484			
2485			
2486			
2487			
2488			
2489			
2490			
2491			
2492			
2493			
2494			
2495			
2496			
2497			
2498			
2499			
2500			

2465
2464
2463
2462
2461
2460
2459
2458
2457
2456
2455
2454
2453
2452
2451
2450
2449
2448
2447
2446
2445
2444
2443
2442
2441
2440
2439
2438
2437
2436
2435
2434
2433
2432
2431
2430
2429
2428
2427
2426
2425
2424
2423
2422
2421
2420
2419
2418
2417
2416
2415
2414
2413
2412
2411
2410
2409
2408
2407
2406
2405
2404
2403
2402
2401
2400
2399
2398
2397
2396
2395
2394
2393
2392
2391
2390
2389
2388
2387
2386
2385
2384
2383
2382
2381
2380
2379
2378
2377
2376
2375
2374
2373
2372
2371
2370
2369
2368
2367
2366
2365
2364
2363
2362
2361
2360
2359
2358
2357
2356
2355
2354
2353
2352
2351
2350
2349
2348
2347
2346
2345
2344
2343
2342
2341
2340
2339
2338
2337
2336
2335
2334
2333
2332
2331
2330
2329
2328
2327
2326
2325
2324
2323
2322
2321
2320
2319
2318
2317
2316
2315
2314
2313
2312
2311
2310
2309
2308
2307
2306
2305
2304
2303
2302
2301
2300
2299
2298
2297
2296
2295
2294
2293
2292
2291
2290
2289
2288
2287
2286
2285
2284
2283
2282
2281
2280
2279
2278
2277
2276
2275
2274
2273
2272
2271
2270
2269
2268
2267
2266
2265
2264
2263
2262
2261
2260
2259
2258
2257
2256
2255
2254
2253
2252
2251
2250
2249
2248
2247
2246
2245
2244
2243
2242
2241
2240
2239
2238
2237
2236
2235
2234
2233
2232
2231
2230
2229
2228
2227
2226
2225
2224
2223
2222
2221
2220
2219
2218
2217
2216
2215
2214
2213
2212
2211
2210
2209
2208
2207
2206
2205
2204
2203
2202
2201
2200
2199
2198
2197
2196
2195
2194
2193
2192
2191
2190
2189
2188
2187
2186
2185
2184
2183
2182
2181
2180
2179
2178
2177
2176
2175
2174
2173
2172
2171
2170
2169
2168
2167
2166
2165
2164
2163
2162
2161
2160
2159
2158
2157
2156
2155
2154
2153
2152
2151
2150
2149
2148
2147
2146
2145
2144
2143
2142
2141
2140
2139
2138
2137
2136
2135
2134
2133
2132
2131
2130
2129
2128
2127
2126
2125
2124
2123
2122
2121
2120
2119
2118
2117
2116
2115
2114
2113
2112
2111
2110
2109
2108
2107
2106
2105
2104
2103
2102
2101
2100
2099
2098
2097
2096
2095
2094
2093
2092
2091
2090
2089
2088
2087
2086
2085
2084
2083
2082
2081
2080
2079
2078
2077
2076
2075
2074
2073
2072
2071
2070
2069
2068
2067
2066
2065
2064
2063
2062
2061
2060
2059
2058
2057
2056
2055
2054
2053
2052
2051
2050
2049
2048
2047
2046
2045
2044
2043
2042
2041
2040
2039
2038
2037
2036
2035
2034
2033
2032
2031
2030
2029
2028
2027
2026
2025
2024
2023
2022
2021
2020
2019
2018
2017
2016
2015
2014
2013
2012
2011
2010
2009
2008
2007
2006
2005
2004
2003
2002
2001
2000

.SBTTL TRAP CATCHER

000000

.=0
;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS

.SBTTL STARTING ADDRESS(ES)

000200

.=200

000200 000137 001202

JMP @*START ;; JUMP TO STARTING ADDRESS OF PROGRAM
;*****

.SBTTL ACT11 HOOKS

;*THE FOLLOWING LOCATIONS ARE SETUP TO BE USED WITH ACT11
;* *
;*LOCATION 46 WILL CONTAIN THE ADDRESS OF THE LOGICAL
;*END OF THE PROGRAM.
;*LOCATION 52 IS USED TO SPECIFY PROGRAM OPERATING REQUIREMENTS
;*AND/OR RESTRICTIONS. THIS IS ACCOMPLISHED BY SETTING VARIOUS BITS
;*TO A ONE OR A ZERO. THE BITS USED AND THERE MEANING ARE:

BIT 15=1 PROGRAM SHOULD BE POWER FAILED WHILE RUNNING
=0 NO POWER FAIL DESIRED

BIT 14=1 PROGRAM RUN TIME IS MEMORY SIZE DEPENDENT
=0 RUN TIME IS NOT MEMORY SIZE DEPENDENT

BITS 13-0 MUST BE ZERO'S

000204
000046
000046 010732
000052
000052 000000
000204

SSVPC=. ;; SAVE LOCATION COUNTER
. =46 ;; SET LOCATION COUNTER
.WORD SENDAD ;; SET LOC.46 TO ADDRESS SENDAD
. =52 ;; SET LOCATION COUNTER
.WORD 0 ;; SET LOC.52 TO ZERO
. =SSVPC ;; RESTORE LOCATION COUNTER

2466
2467
2468
2469
2470
2471
2472
2473
2474
2475 001100
2476 001100 000000
2477 001102 000
2478 001103 000
2479 001104 000000
2480 001106 000000
2481 001110 000000
2482 001112 000000
2483 001114 000
2484 001115 001
2485 001116 000000
2486 001120 000000
2487 001122 000000
2488 001124 000000
2489 001126 000000
2490 001130 000000 000000 000000
2491 001136 177560
2492 001140 177562
2493 001142 177564
2494 001144 177566
2495 001146 000
2496 001147 002
2497 001150 012
2498 001151 000
2499 001152 000000
2500
2501 001154 000000
2502 001156 000000
2503 001160 000000
2504 001162 000000
2505 001164 000000
2506 001166 000000
2507 001170 000000
2508 001172 077
2509 001173 015
2510 001174 000012
2511 001176
2512 001176 000000
2513 001200 000000

;;*****

.SBTTL COMMON TAGS

;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
;*USED IN THE PROGRAM.

.=1100

\$CMTAG: .WORD 0
\$PASS: .WORD 0
\$STNM: .BYTE 00
\$ERFLG: .BYTE 00
\$ICNT: .WORD 00
\$LPADR: .WORD 00
\$LPERR: .WORD 00
\$ERTTL: .WORD 00
\$ITEMB: .BYTE 0
\$ERMAX: .BYTE 1
\$ERRPC: .WORD 0
\$GDADR: .WORD 0
\$BDADR: .WORD 0
\$GDDAT: .WORD 0
\$BDDAT: .WORD 0,0,0
\$TKS: 177560
\$TKB: 177562
\$TPS: 177564
\$TPB: 177566
\$NULL: .BYTE 0
\$FILLS: .BYTE 2
\$FILLC: .BYTE 12
\$TPFLG: .BYTE 0
\$REGAD: .WORD 0
\$REG0: .WORD 0
\$REG1: .WORD 00
\$REG2: .WORD 00
\$STMP0: .WORD 00
\$STMP1: .WORD 00
\$STMP2: .WORD 00
\$STMP3: .WORD 0
\$QUES: .ASCII ??
\$CRLF: .ASCII <15>
\$LF: .ASCII <12>
\$ERPSW: .WORD 0

;; START OF COMMON TAGS
;; CONTAINS PASS COUNT
;; CONTAINS THE TEST NUMBER
;; CONTAINS ERROR FLAG
;; CONTAINS SUBTEST ITERATION COUNT
;; CONTAINS SCOPE LOOP
;; CONTAINS SCOPE RETURN FOR ERRORS
;; CONTAINS TOTAL ERRORS DETECTED
;; CONTAINS ITEM CONTROL BYTE
;; CONTAINS MAX. ERRORS PER TEST
;; CONTAINS PC OF LAST ERROR INSTRUCTION
;; CONTAINS OF 'GOOD' DATA
;; CONTAINS OF 'BAD' DATA
;; CONTAINS 'GOOD' DATA
;; CONTAINS 'BAD' DATA
;; RESERVED--NOT TO BE USED
;; TTY KBD STATUS
;; TTY KBD BUFFER
;; TTY PRINTER STATUS REG.
;; TTY PRINTER BUFFER REG.
;; CONTAINS NULL CHARACTER FOR FILLS
;; CONTAINS # OF FILLER CHARACTERS REQUIRED
;; INSERT FILL CHARS. AFTER A "LINE FEED"
;; "TERMINAL AVAILABLE" FLAG (BIT:07)=0=YES)
;; CONTAINS THE FROM
;; WHICH (\$REG0) WAS OBTAINED
;; CONTAINS ((\$REGAD)+0)
;; CONTAINS ((\$REGAD)+2)
;; CONTAINS ((\$REGAD)+4)
;; USER DEFINED
;; USER DEFINED
;; USER DEFINED
;; USER DEFINED
;; USER DEFINED
;; QUESTION MARK
;; CARRIAGE RETURN
;; LINE FEED

2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569

001202

.SBTTL ERROR POINTER TABLE

;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
;*LOCATION \$ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
;*NOTE1: IF \$ITEMB IS 0 THE ONLY PERTINENT DATA IS (\$ERRPC).
;*NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

;* EM ::POINTS TO THE ERROR MESSAGE
;* DH ::POINTS TO THE DATA HEADER
;* DT ::POINTS TO THE DATA
;* DF ::POINTS TO THE DATA FORMAT

\$ERRTB:

FOLLOWING IS AN INDEX OF THE POSSIBLE FAILURES THAT
CAUSE A TRAP TO LOCATIONS 4, 10, 14, OR 24 OR THAT
CAUSE THE PROCESSOR TO HANG (H). NGT=NOT GETTING THRU

TEST NUMBER	EFFECT	CAUSE
25	10	RACH NEG.B*DMO NOT GOING HIGH
25	24	RACH A2 RAB00 NOT GOING LOW
25	H	RACH E59(6) NOT GOING LOW OR NGT RACL RAD07
26	H	RACL RAD00 NOT GOING LOW
26	4	EITHER IRCC SM357 STUCK H OR RACL E70 BAD
26	14	IRCC C0 RAB03 NOT GOING H OR RACL RAD03 INPUT STUCK LOW
26	4	SRC CONST ADDED 1 OR 3
27	H	RACK RAD01 NOT GOING LOW
31	10	RACK A0 RAB01 NOT GOING LOW OR NGT RACL RAD01
31	4	RACK BRAB05 NOT GOING LOW OR NGT PACL RAD05
31	4	1\$ ON TOP OF STACK
32	10	DST CONST ADDED 1 OR 3
33	10	RACE A0 RAB02 DOES NOT GO LOW
34	10	RACE E44 IS BAD
34	10	IRCB K/CLASS STUCK LOW
34	H	GRAB 0B0(1) STUCK H OR NGT RACL E71
34	H	GRAD DRMX00 STUCK H OR GRAB E50 BAD
35	10	RACE BIN*SMO H DID NOT GO HIGH
35	10	IR DECODE ROM WORD BAD
36	10	RACE BIN*SMC FAILED

2570	*	36	10	IR DECODE ROM WORD BAD
2571	*			
2572	*	37	10	RACE E45 BAD
2573	*			
2574	*	40	10	RACE AD RAB00 DOES NOT GO LOW
2575	*	40	10	IRCB(JMP+JSR) IS STUCK LOW
2576	*	40	H	IRCB FJ CLASS IS STUCK HIGH
2577	*			
2578	*	41	10	RACE E45 IS BAD
2579	*			
2580	*	42	10	RACE E33 IS BAD
2581	*			
2582	*	43	10	RACH U/CLASS NOT GOING HIGH
2583	*			SS-2 ON TOP OF STACK
2584	*	43	10	EITHER RACE E10 BAD OR
2585	*			RACE BIN NOT GOING H
2586	*			SS ON TOP OF STACK
2587	*			
2588	*	44	10	RACJ AFIR 14(1) NGT RACE E42
2589	*	44	10	IRCB E38(6) STUCK HIGH
2590	*			
2591	*	45	H	GRAB OBD(0) STUCK H OR NGT RACK E51
2592	*			
2593	*	46	10	RACE E33 BAD
2594	*			
2595	*	54	10	RACF E3 DOES NOT GO HIGH
2596	*			
2597	*	56	10	PART PCLASS FIELD BAD IN IR DECODE ROM
2598	*			
2599	*	60	10	PART PCLASS FIELD BAD IN IR DECODE ROM
2600	*			
2601	*	62	10	IRCC CD RAB03 STUCK H OR NGT RACL RAD03
2602	*			OR FORK C MUX INPUT B0 STUCK H
2603	*			
2604	*	65	10	B FORK MUX SELECT STUCK LOW
2605	*	65	H	IRCB B0 RAB04 NGT RAD05
2606	*	65	H	B FORK MUX INPUT B0 STUCK H OR
2607	*			IRCC B0 RAB00 STUCK H
2608	*	65	4	B FORK MUX INPUT B3 OR
2609	*			IRCB B0 RAB00 STUCK L
2610	*	65	H	IRCB E46(12) STUCK L-MICRO ADR 170
2611	*			
2612	*	67	10	RACE E35(1) BAD
2613	*			
2614	*	70	10	B FORK MUX STROBE STUCK L (CHIP FAILURE)
2615	*			
2616	*	75	10	RACE JMP+JSR+SWAB NOT GOING HIGH
2617	*	75	10	IRCB E63 BAD-R5 CONTAINS "T67+2"
2618	*			
2619	*	105	4	RACF (HALT:OP CODE 7) DOES NOT GO HIGH
2620	*	105	4	RACE E7 BAD-ODD ADR BIT SET IN ERROR REG

 * THE FOLLOWING FIVE CONDITION CODE AND BRANCH TESTS ARE A
 * FUNCTION OF RACK F TRUE 1. SECTION 1 OF EACH TEST IS
 * DEPENDENT ON TRUE 1 NOT GOING HIGH, WHILE SECTION 2 IS
 *

M05

```
2626 ;* DEPENDENT ON TRUE ONE GOING HIGH.
2627 ;*
2628 001202 012737 000014 177746 START: MOV #14,0#CONTRL ;FORCE MISSES IN CACHE
2629 ;*****
2630 ;*TEST 1 CCC*BRANCH THRU FET.13
2631 ;*
2632 ;* THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TRUE 1:
2633 ;* SECTION 1
2634 ;* BCS E58(13,12) L,H
2635 ;* BMI E59(10,11,9) H,L,H
2636 ;* BVS E48(5,3,4) H,L,H
2637 ;* BLOS E59(4,3,5) H,L,H
2638 ;*****
2639 001210 000257 TST1: CCC ;CC=0000
2640 ;SECTION 1
2641 001212 103404 BCS 1$
2642 001214 100403 BMI 1$
2643 001216 102402 BVS 1$
2644 001220 101401 BLOS 1$
2645 001222 103001 BCC TST2 ;;GO TO NEXT TEST
2646 001224
2647 001224 000000 1$: HALT ;RACF TRUE 1 WENT HIGH OR
2648 ;RACH A2 RAB02 IS NOT GOING HIGH
2649 ;OR NOT GETTING THRU RACL RAD02
2650 ;FOR LOOPING CHANGE TO "BR TST1" (771)
2651 ;*****
2652 ;*TEST 2 SEC*BRANCH THRU FET.13 AND FET.11
2653 ;*
2654 ;* THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TRUE 1:
2655 ;* SECTION 1
2656 ;* BMI E58(13,12) H,L
2657 ;* BVS E58(13,12) H,L
2658 ;* SECTION 2
2659 ;* BCC E58(13,12) H,H
2660 ;*****
2661 001226 000261 TST2: SEC ;CC=0001
2662 ;SECTION 1
2663 001230 100402 BMI 1$
2664 001232 102401 BVS 1$
2665 001234 100001 BPL 2$ ;GO TO SECTION 2
2666 001236
2667 001236 000000 1$: HALT ;RACF E58 FAILED
2668 ;FOR LOOPING CHANGE TO "BR TST2" (773)
2669 ;SECTION 2
2670 001240 103001 2$: BCC 3$
2671 001242 103401 BCS TST3 ;;GO TO NEXT TEST
2672 001244
2673 001244 000000 3$: HALT ;EITHER RACF TRUE 1 DID NOT GO HIGH
2674 ;OR IT DID NOT GET THRU RACH A2 RAB00
2675 ;FOR LOOPING CHANGE TO "BR 2$" (775)
2676 ;*****
2677 ;*TEST 3 SEV*BRANCH THRU FET.13 AND FET.11
2678 ;*
2679 ;* THE FOLLOWING IS A LIST OF PATTERNS PUT ON THE GATES OF TRUE 1:
2680 ;* SECTION 1
2681 ;* BCS E48(5,3,4) L,H,H
```

N05

```

2682          ;*          BMI      E48(5,3,4)      H,H,L
2683          ;*          SECTION 2
2684          ;*          BVC      E48(5,3,4)      H,H,H
2685          ;*****
2686 001246 000257 TST3: CCC          ;CC=0000
2687          ;SECTION 1
2688 001250 000262          SEV          ;CC=0010
2689 001252 103402          BCS      1$
2690 001254 100401          BMI      1$
2691 001256 100001          BPL      2$
2692 001260          1$:
2693 001260 000000          HALT          ;RACF E48 FAILED
2694          ;FOR LOOPING CHANGE TO "BR TST3" (772)
2695          ;SECTION 2
2696 001262 102001 2$: BVC      3$
2697 001264 102401          BVS      TST4          ;;GO TO NEXT TEST
2698 001266
2699 001266 000000 3$: HALT          ;EITHER RACF E48 OR E47(1) FAILED
2700          ;FOR LOOPING CHANGE TO "BR 2$" (775)
2701          ;*****
2702          ;*TEST 4          SEZ*BRANCH THRU FET.13 AND FET.11
2703          ;*
2704          ;*          THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TRUE 1:
2705          ;*          SECTION 1
2706          ;*          BCS      E59(4,3,5)      H,H,L
2707          ;*          BMI      E59(4,3,5)      L,H,H
2708          ;*          SECTION 2
2709          ;*          BHI      E59(4,3,5)      H,H,H
2710          ;*****
2711 001270 000257 TST4: CCC          ;CC=0000
2712          ;SECTION
2713 001272 000264          SEZ          ;CC=0100
2714 001274 103402          BCS      1$
2715 001276 100401          BMI      1$
2716 001300 102001          BVC      2$          ;GO TO SECTION 2
2717 001302
2718 001302 000000 1$: HALT          ;RACF E59 FAILED
2719          ;FOR LOOPING CHANGE TO "BR TST4" (772)
2720          ;SECTION 2
2721 001304 101001 2$: BHI      3$
2722 001306 101401          BLOS     TST5          ;;GO TO NEXT TEST
2723 001310
2724 001310 000000 3$: HALT          ;EITHER RACF E59 OR E47(11) FAILED
2725          ;FOR LOOPING CHANGE TO "BR 2$" (775)
2726          ;*****
2727          ;*TEST 5          SEN*BRANCH THRU FET.13 AND FET.11
2728          ;*
2729          ;*          THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TRUE 1:
2730          ;*          SECTION 1
2731          ;*          BLOS     E59(10,11,9)     H,H,L
2732          ;*          BVS      E59(10,11,9)     L,H,H
2733          ;*          SECTION 2
2734          ;*          BPL      E59(10,11,9)     H,H,H
2735          ;*****
2736 001312 000257 TST5: CCC          ;CC=0000
2737          ;SECTION 1

```

2738 001314 000270
2739 001316 101402
2740 001320 102401
2741 001322 101001
2742 001324
2743 001324 000000
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765 001334 000257
2766 001336 003403
2767 001340 002402
2768 001342 001401
2769 001344 003001
2770 001346
2771 001346 000000
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782 001350 000257
2783
2784 001352 000262
2785 001354 001401
2786 001356 001001
2787 001360
2788 001360 000000
2789
2790
2791 001362 003001
2792 001364 003401
2793 001366

```
SEV                                     ;CC=1000
BL0S                                   1S
BVS                                    1S
BHI                                    2S
;GO TO NEXT SECTION
1S:  HALT                               ;RACF E59 FAILED
                                       ;FOR LOOPING CHANGE TO "BR TST5" (772)
;SECTION 2
2S:  BPL                                3S
     BMI                                TST6
;GO TO NEXT TEST
3S:  HALT                               ;EITHER RACF E59 OR E47(13) FAILED
                                       ;FOR LOOPING CHANGE TO "BR 2S" (775)
;*****
;* THE FOLLOWING SEVEN TESTS ARE A FUNCTION OF RACF TRUE 2.
;* SECTION 1 OF EACH TEST IS DEPENDENT ON TRUE 2 NOT GOING HIGH
;* WHILE SECTION 2 IS DEPENDENT ON TRUE 2 GOING HIGH.
;*****
;* TEST 6 BRANCHES THRU FET.13
;*
;* THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TRUE 2:
;* SECTION 1
;* BLE E58(2,1)H,L E59(13,1,2)L,H,H E48(1,13,2)H,H,L
;* BLT E59(13,1,2)L,H,H E48(1,13,2)H,H,L E58(10,9)L,H
;* BEQ E58(2,1)H,L E58(10,9)H,L
;*****
;* TST6: CCC ;CC=0000
;* BLE 1S
;* BLT 1S
;* BEQ 1S
;* BGT TST7 ;GO TO NEXT TEST
1S:  HALT                               ;RACF TRUE 2 WENT HIGH
                                       ;FOR LOOPING CHANGE TO "BR TST6" (772)
;*****
;* TEST 7 BRANCH THRU FET.13 AND FET.12
;*
;* THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TRUE 2:
;* SECTION 1
;* BEQ E48(1,13,2) L,H,H
;* SECTION 2
;* BGT E48(1,13,2) H,H,H
;*****
;* TST7: CCC ;CC=0000
;* SECTION 1
;* SEV ;CC=0010
;* BEQ 1S
;* BNE 2S ;GO TO SECTION 2
1S:  HALT                               ;RACF E48 FAILED
                                       ;FOR LOOPING CHANGE TO "BR TST7" (773)
;SECTION 2
2S:  BGT 3S
     BLE TST10 ;GO TO NEXT TEST
3S:
```

2794 001366 000000
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806 001370 000257
2807
2808 001372 000264
2809 001374 002401
2810 001376 002001
2811 001400
2812 001400 000000
2813
2814
2815 001402 001001
2816 001404 001401
2817 001406
2818 001406 000000
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829 001410 000257
2830
2831 001412 000270
2832 001414 001401
2833 001416 001001
2834 001420
2835 001420 000000
2836
2837
2838 001422 002001
2839 001424 002401
2840 001426
2841 001426 000000
2842
2843
2844
2845
2846
2847
2848
2849

```
HALT ; EITHER RACF TRUE 2 DID NOT GO HIGH  
; OR IT DID NOT GET THRU RACH A2 RAB01  
; FOR LOOPING CHANGE TO "BR 25" (775)  
*****  
*TEST 10 BRANCH THRU FET.13 AND FET.12  
*  
* THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TRUE 2:  
* SECTION 1  
* BLT E58(2,1) L,H  
* SECTION 2  
* BNE E59(2,1) H,H  
*****  
TST10: CCC ;CC=0000  
;SECTION 1  
SEZ ;CC=0100  
BLT 1$  
BGE 2$ ;GO TO SECTION 2  
1$: HALT ;RACF E58 FAILED  
;FOR LOOPING CHANGE TO "BR TST10" (773)  
SECTION 2  
2$: BNE 3$  
BEQ TST11 ;;GO TO NEXT TEST  
3$: HALT ; EITHER RACF E58 OR E46(12) FAILED  
;FOR LOOPING CHANGE TO "BR 25" (775)  
*****  
*TEST 11 BRANCH THRU FET.13 AND FET.12  
*  
* THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TRUE 2:  
* SECTION 1  
* BEQ E59(13,1,2) H,H,L  
* SECTION 2  
* BGE E59(13,1,2) H,H,H  
*****  
TST11: CCC ;CC=0000  
;SECTION 1  
SEN ;CC=1000  
BEQ 1$  
BNE 2$ ;GO TO NEXT SECTION  
1$: HALT ;RACF E59 FAILED  
;FOR LOOPING CHANGE TO "BR TST11" (773)  
SECTION 2  
2$: BGE 3$  
BLT TST12 ;;GO TO NEXT TEST  
3$: HALT ; EITHER RACF E59 OR E46(13) FAILED  
;FOR LOOPING CHANGE TO "BR 25" (775)  
*****  
*TEST 12 BRANCHES THRU FET.13 AND FET.12  
*  
* THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TRUE 2:  
* SECTION 1  
* BEQ E58(2,1)H,L E58(10,9)H,L  
* BLT E59(13,1,2)H,L,H E48(1,13,2)H,L,H E58(10,9)L,H
```

2850
2851 001430 000262
2852
2853 001432 001402
2854 001434 002401
2855 001436 001001
2856 001440
2857 001440 000000
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869 001442 000277
2870 001444 000244
2871 001446 005000
2872 001450 103403
2873 001452 102402
2874 001454 100401
2875 001456 001401
2876 001460
2877 001460 000000
2878
2879
2880 001462 005000
2881 001464 000277
2882 001466 000244
2883 001470 005700
2884 001472 103403
2885 001474 102402
2886 001476 100401
2887 001500 001401
2888 001502
2889 001502 000000
2890
2891
2892 001504 005000
2893 001506 000257
2894 001510 000266
2895 001512 005100
2896 001514 100003
2897 001516 001402
2898 001520 102401
2899 001522 103401
2900 001524
2901 001524 000000
2902
2903
2904 001526 005000
2905 001530 000277

```
*****
TST12: SEV ;CC=1010
:SECTION 1
      BEQ 15
      BLT 15
      BNE TST13 ;;GO TO NEXT TEST
IS: HALT ;RACF TRUE 2 WENT HIGH
;FOR LOOPING CHANGE TO "BR TST12" (773)
*****
:TEST 13 UNIARY AND BINARY (SMD)
*
* THE FOLLOWING TEST TESTS ALL THE E/CLASS
* INSTRUCTIONS WITH A DESTINATION MODE OF 0
* AND DESTINATION FIELD OF NOT 7. THIS CLASS CONSISTS
* OF ALL THE UNIARY (EXCEPT NEG) INSTRUCTIONS AND
* ALL THE BINARY INSTRUCTIONS WITH A SOURCE MODE
* OF 0.
*****
TST13: SCC ;CC'S=1011
      CLZ ;RO=000000, CC'S=0100
      C'R RO
      BCS CLRRO
      BVS CLRRO
      BMI CLRRO
      BEQ .+4
CLRRO: HALT ;ERROR, INCORRECT CC'S AFTER CLR
;FOR LOOPING CHANGE TO "BR TST13" (770)
      CLR RO
      SCC ;CC'S=1111
      CLZ ;RO=000000, CC'S=1011
      TST RO ;RO=000000, CC'S=0100
      BCS TSTRO
      BVS TSTRO
      BMI TSTRO
      BEQ .+4
TSTRO: HALT ;ERROR, INCORRECT CC'S AFTER TST
;FOR LOOPING CHANGE TO "BR CLRRO+2" (767)
      CLR RO
      CCC ;CC'S=0000
      +SEZ:SEV ;RO=000000, CC'S=0110
      COM RO ;RO=177777, CC'S=1001
      BPL COMRO
      BEQ COMRO
      BVS COMRO
      BCS .+4
COMRO: HALT ;ERROR, INCORRECT CC'S AFTER COM
;FOR LOOPING CHANGE TO "BR TSTRO+2" (767)
      CLR RO
      SCC ;RO=000000, CC'S=1111
```

2906	001532	005500	AUC	RO	;RO=000001, CC'S=0000
2907	001534	100403	BMI	ADCRO	
2908	001536	001402	BEQ	ADCRO	
2909	001540	102401	BVS	ADCRO	
2910	001542	103001	BCC	.+4	
2911	001544		ADCRO:		
2912	001544	000000	HALT		;ERROR, INCORRECT CC'S AFTER ADC ;FOR LOOPING CHANGE TO "BR COMRO+2" (770)
2913					
2914					
2915	001546	005000	CLR	RO	
2916	001550	000261	SEC		
2917	001552	005500	ADC	RO	
2918	001554	000257	CCC		
2919	001556	000270	SEN		;RO=000001, CC'S=1000
2920	001560	006000	ROR	RO	;RO=000000, CC'S=0111
2921	001562	100403	BMI	RORRO	
2922	001564	001002	BNE	RORRO	
2923	001566	102001	BVC	RORRO	
2924	001570	103401	BCS	.+4	
2925	001572		RORRO:		
2926	001572	000000	HALT		;ERROR, INCORRECT CC'S AFTER ROR ;FOR LOOPING CHANGE TO "BR ADCRO+2" (765)
2927					
2928					
2929	001574	005000	CLR	RO	
2930	001576	000277	SCC		
2931	001600	000250	CLN		;RO=000000, CC'S=0111
2932	001602	005300	DEC	RO	;RO=177777, CC'S=1001
2933	001604	100003	BPL	DECRO	
2934	001606	001402	BEQ	DECRO	
2935	001610	102401	BVS	DECRO	
2936	001612	103401	BCS	.+4	
2937	001614		DECRO:		
2938	001614	000000	HALT		;ERROR, INCORRECT CC'S AFTER DEC ;FOR LOOPING CHANGE TO "BR RORRO+2" (767)
2939					
2940					
2941	001616	005000	CLR	RO	
2942	001620	000277	SCC		;RO=000000, CC'S=1111
2943	001622	005200	INC	RO	;RO=000001, CC'S=0000
2944	001624	100403	BMI	INCRO	
2945	001626	001402	BEQ	INCRO	
2946	001630	102401	BVS	INCRO	
2947	001632	103401	BCS	.+4	
2948	001634		INCRO:		
2949	001634	000000	HALT		;ERROR, INCORRECT CC'S AFTER INC ;FOR LOOPING CHANGE TO "BR DECRO+2" (770)
2950					
2951					
2952	001636	005000	CLR	RO	
2953	001640	000277	SCC		
2954	001642	000244	CLZ		;RO=000000, CC'S=1011
2955	001644	006300	ASL	RO	;RO=000000, CC'S=0100
2956	001646	100403	BMI	ASLRO	
2957	001650	001002	BNE	ASLRO	
2958	001652	102401	BVS	ASLRO	
2959	001654	103001	BCC	.+4	
2960	001656		ASLRO:		
2961	001656	000000	HALT		;ERROR, INCORRECT CC'S AFTER ASL

```

2962                                     ;FOR LOOPING CHANGE TO "BR INCRD+2" (767)
2963
2964 001660 005000          CLR      RO
2965 001662 000261          SEC
2966 001664 006000          ROR      RO          ;RO=100000, CC'S=1000
2967 001666 006100          ROL      RO          ;RO=000000, CC'S=0111
2968 001670 100403          BMI      ROLRO
2969 001672 001002          BNE      ROLRO
2970 001674 102001          BVC      ROLRO
2971 001676 103401          BCS      .+4
2972                                     ROLRO:
2973 001700 000000          HALT
2974                                     ;ERROR, INCORRECT CC'S AFTER ROL
2975                                     ;FOR LOOPING CHANGE TO "BR ASLRO+2" (767)
2976
2976 001702 005000          CLR      RO
2977 001704 000261          SEC
2978 001706 006000          ROR      RO
2979 001710 005200          INC      RO
2980 001712 000277          SCC
2981 001714 000251          +CLN!CLC
2982 001716 006200          ASR      RO          ;RO=100000, CC'S=0110
2983 001720 100003          BPL      ASRRO
2984 001722 001402          BEQ      ASRRO
2985 001724 102401          BVS      ASRRO
2986 001726 103401          BCS      .+4
2987                                     ASRRO:
2988 001730 000000          HALT
2989                                     ;ERROR, INCORRECT CC'S AFTER ASR
2990                                     ;FOR LOOPING CHANGE TO "BR RORRO+2" (721)
2991
2991 001732 005000          CLR      RO
2992 001734 000277          SCC
2993 001736 000250          CLN
2994 001740 005600          SBC      RO          ;RO=000000, CC'S=0111
2995 001742 100003          BPL      SBCRO
2996 001744 001402          BEQ      SBCRO
2997 001746 102401          BVS      SBCRO
2998 001750 103401          BCS      .+4
2999                                     SBCRO:
3000 001752 000000          HALT
3001                                     ;ERROR, INCORRECT CC'S AFTER SBC
3002                                     ;FOR LOOPING CHANGE TO "BR ASRRO+2" (767)
3003
3003 001754 005000          CLR      RO
3004 001756 000261          SEC
3005 001760 006000          ROR      RO
3006 001762 000277          SCC
3007 001764 000250          CLN
3008 001766 000300          SWAB      RO          ;RO=100000, CC'S=0111
3009 001770 100003          BPL      SWABRO
3010 001772 001402          BEQ      SWABRO
3011 001774 102401          BVS      SWABRO
3012 001776 103001          BCC      .+4
3013                                     SWABRO:
3014 002000 000000          HALT
3015                                     ;ERROR, INCORRECT CC'S AFTER SWAB
3016                                     ;FOR LOOPING CHANGE TO "BR SBCRO+2" (765)
3017
3017 002002 005000          CLR      RO
    
```

3018	002004	005300	DEC	RO	
3019	002006	000257	CCC		
3020	002010	000262	SEV		;RO=177777, CC'S=0010
3021	002012	006700	SXT	RO	;RO=000000, CC'S=0100
3022	002014	100403	BMI	SXTRO	
3023	002016	001002	BNE	SXTRO	
3024	002020	102401	BVS	SXTRO	
3025	002022	103001	BCC	.+4	
3026	002024		SXTRO:		
3027	002024	000000	HALT		;ERROR, INCORRECT CC'S AFTER SXT
3028					;FOR LOOPING CHANGE TO "BR SWABRO+2" (766)
3029	002026	005000	CLR	RO	
3030	002030	000270	SEN		;RO=000000, CC'S=1XXX
3031	002032	006700	SXT	RO	;RO=177777, CC'S=1XXXX
3032	002034	005200	INC	RO	
3033	002036	001401	BEQ	.+4	
3034	002040		SXT2:		
3035	002040	000000	HALT		;SIGN EXTEND FAILED WITH N SET
3036					;FOR LOOPING CHANGE TO "BR SXTRO+2" (772)
3037					
3038	002042	005000	CLR	RO	
3039	002044	005300	DEC	RO	
3040	002046	000277	SCC		
3041	002050	000244	CLZ		;RO=177777, CC'S=1011
3042	002052	074000	XOR	RO,RO	;RO=000000, CC'S=0101
3043	002054	100403	BMI	XORRO	
3044	002056	001002	BNE	XORRO	
3045	002060	102401	BVS	XORRO	
3046	002062	103401	BCS	.+4	
3047	002064		XORRO:		
3048	002064	000000	HALT		;ERROR, INCORRECT CC'S AFTER XOR
3049					;FOR LOOPING CHANGE TO "BR SXT2+2" (766)
3050					
3051	002066	005000	CLR	RO	
3052	002070	000261	SEC		
3053	002072	006000	ROR	RO	
3054	002074	000300	SWAB	RO	
3055	002076	000277	SCC		
3056	002100	000250	CLN		;RO=000200, CC'S=0111
3057	002102	110000	MOVB	RO,RO	;RO=177600, CC'S=1001
3058	002104	100010	BPL	MOV BRO	
3059	002106	001407	BEQ	MOV BRO	
3060	002110	102406	BVS	MOV BRO	
3061	002112	103005	BCC	MOV BRO	
3062	002114	000250	CLN		
3063	002116	000264	SEZ		;RO=177600, CC'S=0100
3064	002120	005700	TST	RO	;CC'S=1000
3065	002122	100002	BPL	MOVRO	
3066	002124	001002	BNE	.+6	
3067	002126		MOV BRO:		
3068	002126	000000	HALT		;ERROR, INCORRECT CC'S AFTER MOV B
3069					;FOR LOOPING CHANGE TO "BR XORRO+2" (757)
3070	002130		MOVRO:		
3071	002130	000000	HALT		;ERROR, MOV B DID NOT SIGN EXTEND
3072					;FOR LOOPING CHANGE TO "BR XORRO+2" (756)
3073					

3074	002132	005000	CLR	RO	
3075	002134	000277	SCC		
3076	002136	000244	CLZ		;RO=000000, CC'S=1011
3077	002140	030000	BIT	RO,RO	;RO=000000, CC'S=0101
3078	002142	100403	BMI	BITRO	
3079	002144	001002	BNE	BITRO	
3080	002146	102401	BVS	BITRO	
3081	002150	103401	BCS	.+4	
3082	002152		BITRO:		
3083	002152	000000	HALT		;ERROR, INCORRECT CC'S AFTER BIT
3084					;FOR LOOPING CHANGE TO "BR MOVRO+2" (767)
3085					
3086	002154	005000	CLR	RO	
3087	002156	005200	INC	RO	
3088	002160	000277	SCC		
3089	002162	000244	CLZ		;RO=000001, CC'S=1011
3090	002164	040000	BIC	RO,RO	;RO=000000, CC'S=0101
3091	002166	100403	BMI	BICRO	
3092	002170	001002	BNE	BICRO	
3093	002172	102401	BVS	BICRO	
3094	002174	103401	BCS	.+4	
3095	002176		BICRO:		
3096	002176	000000	HALT		;ERROR, INCORRECT CC'S AFTER BIC
3097					;FOR LOOPING CHANGE TO "BR BITRO+2" (766)
3098					
3099	002200	005000	CLR	RO	
3100	002202	005200	INC	RO	
3101	002204	000277	SCC		;RO=000001, CC'S=1111
3102	002206	050000	BIS	RO,RO	;RO=000001, CC'S=0001
3103	002210	100403	BMI	BISRO	
3104	002212	001402	BEQ	BISRO	
3105	002214	102401	BVS	BISRO	
3106	002216	103401	BCS	.+4	
3107	002220		BISRO:		
3108	002220	000000	HALT		;ERROR, INCORRECT CC'S AFTER BIS
3109					;FOR LOOPING CHANGE TO "BR BICRO+2" (767)
3110					
3111	002222	005000	CLR	RO	
3112	002224	000261	SEC		
3113	002226	006000	ROR	RO	
3114	002230	006000	ROR	RO	
3115	002232	000277	SCC		
3116	002234	000252	+CLN!CLV		;RO=040000, CC'S=0101
3117	002236	060000	ADD	RO,RO	;RO=100000, CC'S=1010
3118	002240	100003	BPL	ADDR0	
3119	002242	001402	BEQ	ADDR0	
3120	002244	102001	BVC	ADDR0	
3121	002246	103001	BCC	.+4	
3122	002250		ADDR0:		
3123	002250	000000	HALT		;ERROR, INCORRECT CC'S AFTER ADD
3124					;FOR LOOPING CHANGE TO "BR BISRO+2" (764)
3125					
3126	002252	005000	CLR	RO	
3127	002254	005200	INC	RO	
3128	002256	000277	SCC		
3129	002260	000244	CLZ		;RO=000001, CC'S=1011

```

3130 002262 160000 SUB RO,RO ;RO=000000, CC'S=0100
3131 002264 100403 BMI SUBRO
3132 002266 001002 BNE SUBRO
3133 002270 102401 BVS SUBRO
3134 002272 103001 BCC .+4
3135 SUBRO:
3136 002274 000000 HALT ;ERROR, INCORRECT CC'S AFTER SUB
;FOR LOOPING CHANGE TO "BR ADDR0+2" (766)
3138
3139 002276 005000 CLR RO
3140 002300 000277 SCC
3141 002302 000244 CLZ ;RO=000000, CC'S=1011
3142 002304 020000 CMP RO,RO ;RO=000000, CC'S=0100
3143 002306 100403 BMI CMPRO
3144 002310 001002 BNE CMPRO
3145 002312 102401 BVS CMPRO
3146 002314 103001 BCC .+4
3147 CMPRO:
3148 002316 000000 HALT ;ERROR, INCORRECT CC'S AFTER CMP
;FOR LOOPING CHANGE TO "BR SUBRO+2" (767)
3149
3150
3151
3152
3153 *****
3154 *TEST 14 REGISTER SELECTION TEST
3155 *
3156 * THIS TEST ENSURES THAT THE 6 ADDRESS LINES INTO
3157 * THE GENERAL PURPOSE REGISTERS (GPR) ARE NOT STUCK.
3158 * THE LABELS OF THE ADDRESS LINES ARE:
3159 * GSAX GENERAL SOURCE ADDRESS LINE
3160 * GDAX GENERAL DESTINATION ADDRESS LINE
3161 * WHERE X STANDS FOR LINE 0, 1, OR 2.
3162 * THE CLASSES OF ERRORS DESCRIBED IN THIS TEST
3163 * ARE DEFINED AS FOLLOWS:
3164 * CLASS A=GDAX OK
3165 * GSAX STUCK
3166 * CLASS B=GSAX OK
3167 * GDAX STUCK
3168 * CLASS C=GSAX STUCK
3169 * GDAX STUCK
3170 *****
3170 002320 005000 TST14: CLR RO
3171 002322 005201 INC R1
3172 002324 005700 TST RO ;DID INC AFFECT RO?
3173 002326 001406 BEQ OVER ;BRANCH ON NOT CLASS B OR C
3174 002330 005000 ROUTO: CLR RO
3175 002332 005201 INC R1
3176 002334 010002 MOV RO,R2 ;DID SO REMAIN 0 ON INC R1?
3177 002336 001001 BNE 2$ ;BR IF YES-NOT CLASS B
3178 002340 000000 HALT ;ERROR, CLASS B FAILURE ON GRAO
;FOR LOOPING CHANGE TO "BR ROUTO" (773)
3179
3180 002342 2$:
3181 002342 000000 HALT ;ERROR, CLASS C FAILURE ON GRAO
;FOR LOOPING CHANGE TO "BR ROUTO" (772)
3182
3183 002344 005201 OVER: INC R1
3184 002346 010001 MOV RO,R1 ;INCREMENT S1 AND D1
3185 002350 001401 BEQ GRAIT ;MOVE SO TO S1 AND D1
;BRANCH-GRAO OK

```



```

3298                                     ;FOR LOOPING CHANGE TO "BR TST16" (770)
3299 002606 005100 2S: COM R0
3300 002610 005102 COM R2
3301 002612 020002 CMP R0,R2 ;DID R2 DST SET?
3302 002614 001401 BEQ 3$ ;BRANCH IF YES
3303 002616 000000 HALT ;ERROR, R2 DST STUCK LOW
3304                                     ;FOR LOOPING CHANGE TO "BR TST16" (763)
3305 002620 020200 3S: CMP R2,R0 ;DID R2 SRC SET?
3306 002622 001401 BEQ TST17 ;BRANCH IF YES
3307 002624 000000 HALT ;ERROR, R2 SRC STUCK LOW
3308                                     ;FOR LOOPING CHANGE TO "BR TST16" (760)

```

```

*****
;TEST 17 GPR3 STUCK BIT TEST

```

```

;LOADS GPR3 WITH ZEROS AND ONES AND COMPARES
;R3 SOURCE AND DESTINATION WITH R0.

```

```

*****

```

```

3315 002626 005000 TST17: CLR R0
3316 002630 005003 CLR R3
3317 002632 020300 CMP R3,R0 ;DID R3 SRC CLEAR?
3318 002634 001401 BEQ 1$ ;BRANCH IF YES
3319 002636 000000 HALT ;ERROR, R3 SRC STUCK HIGH
3320                                     ;FOR LOOPING CHANGE TO "BR TST17" (773)
3321 002640 020003 1S: CMP R0,R3 ;DID R3 DST CLEAR?
3322 002642 001401 BEQ 2$ ;BRANCH IF YES
3323 002644 000000 HALT ;ERROR, R3 DST STUCK HIGH
3324                                     ;FOR LOOPING CHANGE TO "BR TST17" (770)

```

```

3325 002646 005100 2S: COM R0
3326 002650 005103 COM R3
3327 002652 020300 CMP R3,R0 ;DID R3 SRC SET TO ALL ONES?
3328 002654 001401 BEQ 3$ ;BRANCH IF YES
3329 002656 000000 HALT ;ERROR, R3 SRC STUCK LOW
3330                                     ;FOR LOOPING CHANGE TO "BR TST17" (763)
3331 002660 020003 3S: CMP R0,R3 ;DID R3 DST SET TO ALL ONES?
3332 002662 001401 BEQ TST20 ;BRANCH IF YES
3333 002664 000000 HALT ;ERROR, R3 DST STUCK LOW
3334                                     ;FOR LOOPING CHANGE TO "BR TST17" (760)

```

```

*****
;TEST 20 GPR4 STUCK BIT TEST

```

```

;LOADS GPR4 WITH ZEROS AND ONES AND COMPARES
;R4 SOURCE AND DESTINATION WITH R0.

```

```

*****

```

```

3341 002666 005000 TST20: CLR R0
3342 002670 005004 CLR R4
3343 002672 020400 CMP R4,R0 ;DID R4 SRC CLEAR?
3344 002674 001401 BEQ 1$ ;BRANCH IF YES
3345 002676 000000 HALT ;ERROR, R4 SRC STUCK HIGH
3346                                     ;FOR LOOPING CHANGE TO "BR TST20" (773)
3347 002700 020004 1S: CMP R0,R4 ;DID R4 DST CLEAR?
3348 002702 001401 BEQ 2$ ;BRANCH IF YES
3349 002704 000000 HALT ;ERROR, R4 DST STUCK HIGH
3350                                     ;FOR LOOPING CHANGE TO "BR TST20" (770)

```

```

3351 002706 005100 2S: COM R0
3352 002710 005104 COM R4
3353 002712 020004 CMP R0,R4 ;DID R4 DST SET?

```

```

3354 002714 001401          BEQ      3$          ;BRANCH IF YES
3355 002716 000000          HALT                    ;ERROR, R4 DST STUCK LOW
3356                                ;FOR LOOPING CHANGE TO "BR TST20" (763)
3357 002720 020400    3$:  CMP      R4,R0          ;DID R4 SRC SET?
3358 002722 001401          BEQ      TST21         ;BRANCH IF YES
3359 002724 000000          HALT                    ;ERROR, R4 SRC STUCK LOW
3360                                ;FOR LOOPING CHANGE TO "BR TST20" (760)
3361                                ;*****
3362                                ;*TEST 21          GPR5 STUCK BIT TEST
3363                                ;*
3364                                ;*          LOADS R5 WITH ZEROS AND ONES AND COMPARES
3365                                ;*          R5 SOURCE AND DESTINATION WITH R0.
3366                                ;*****
3367 002726 005000    TST21: CLR      R0
3368 002730 005005          CLR      R5
3369 002732 020500          CMP      R5,R0          ;DID R5 SRC CLEAR?
3370 002734 001401          BEQ      1$          ;BRANCH IF YES
3371 002736 000000          HALT                    ;ERROR, R5 SRC STUCK HIGH
3372                                ;FOR LOOPING CHANGE TO "BR TST21" (773)
3373 002740 020005    1$:  CMP      R0,R5          ;DID R5 DST CLEAR?
3374 002742 001401          BEQ      2$          ;BRANCH IF YES
3375 002744 000000          HALT                    ;ERROR, R5 DST STUCK HIGH
3376                                ;FOR LOOPING CHANGE TO "BR TST21" (770)
3377 002746 005100    2$:  COM      R0
3378 002750 005105          COM      R5
3379 002752 020005          CMP      R0,R5          ;DID R5 DST SET TO ALL ONES?
3380 002754 001401          BEQ      3$          ;BRANCH IF YES
3381 002756 000000          HALT                    ;ERROR, R5 DST STUCK LOW
3382                                ;FOR LOOPING CHANGE TO "BR TST21" (763)
3383 002760 020500    3$:  CMP      R5,R0          ;DID R5 SRC SET TO ALL ONES?
3384 002762 001401          BEQ      TST22         ;BRANCH IF YES
3385 002764 000000          HALT                    ;ERROR, R5 SRC STUCK LOW
3386                                ;FOR LOOPING CHANGE TO "BR TST21" (760)
3387                                ;*****
3388                                ;*TEST 22          GPR6 STUCK BIT TEST
3389                                ;*
3390                                ;*          LOADS R6 WITH ZEROS AND ONES AND COMPARES
3391                                ;*          R6 SOURCE AND DESTINATION WITH R0.
3392                                ;*****
3393 002766 005000    TST22: CLR      R0
3394 002770 005106          CLR      R6
3395 002772 020006          CMP      R0,R6          ;DID R6 DST CLEAR?
3396 002774 001401          BEQ      1$          ;BRANCH IF YES
3397 002776 000000          HALT                    ;ERROR, R6 DST STUCK HIGH
3398                                ;FOR LOOPING CHANGE TO "BR TST22" (773)
3399 003000 020600    1$:  CMP      R6,R0          ;DID R6 SRC CLEAR?
3400 003002 001401          BEQ      2$          ;BRANCH IF YES
3401 003004 000000          HALT                    ;ERROR, R6 SRC STUCK HIGH
3402                                ;FOR LOOPING CHANGE TO "BR TST22" (770)
3403 003006 005000    2$:  CLR      R0
3404 003010 005006          CLR      R6
3405 003012 020006          CMP      R0,R6          ;DID R6 DST SET?
3406 003014 001401          BEQ      3$          ;BRANCH IF YES
3407 003016 000000          HALT                    ;ERROR, R6 DST STUCK LOW
3408                                ;FOR LOOPING CHANGE TO "BR TST22" (763)
3409 003020 020600    3$:  CMP      R6,R0          ;DID R6 SRC SET?
    
```

```

3410 003022 001401      BEQ     TST23      ;:BRANCH IF YES
3411 003024 000000      HALT                    ;:ERROR, R6 SRC STUCK LOW
3412                                     ;:FOR LOOPING CHANGE TO "BR TST22" (760)
3413 :*****
3414 :*TEST 23      GPR  SHORTED BIT TEST
3415 :*
3416 :*      TEST IF GPR'S 1 THRU 6 HAVE TWO BITS TIED TOGETHER
3417 :*
3418 :*      NOTE:  RO IS CONSIDERED "HARDCORE"
3419 :*****
3420 003026 005000      TST23: CLR     RO      ;:THIS SECTION OF CODE
3421 003030 005200      INC     RO      ;:
3422 003032 006100      ROL     RO      ;:
3423 003034 006100      ROL     RO      ;:
3424 003036 005200      INC     RO      ;:
3425 003040 006100      ROL     RO      ;:
3426 003042 006100      ROL     RO      ;:
3427 003044 005200      INC     RO      ;:
3428 003046 006100      ROL     RO      ;:
3429 003050 006100      ROL     RO      ;:
3430 003052 005200      INC     RO      ;:
3431 003054 006100      ROL     RO      ;:
3432 003056 006100      ROL     RO      ;:
3433 003060 005200      INC     RO      ;:
3434 003062 006100      ROL     RO      ;:
3435 003064 006100      ROL     RO      ;:
3436 003066 005200      INC     RO      ;:
3437 003070 006100      ROL     RO      ;:
3438 003072 006100      ROL     RO      ;:
3439 003074 005200      INC     RO      ;:
3440 003076 006100      ROL     RO      ;:
3441 003100 006100      ROL     RO      ;:
3442 003102 005200      INC     RO      ;:PUTS 52525 IN RO
3443 003104 010001      2$:  MOV     RO,R1  ;:PUT RO SRC IN R1
3444 003106 020001      CMP     RO,R1  ;:IS R1 DST OK?
3445 003110 001401      BEQ     3$     ;:BRANCH IF YES
3446 003112 000000      HALT                    ;:ERROR, R1DST HAS SHORTED BITS
3447                                     ;:FOR LOOPING CHANGE TO "BR TST15" (605)
3448 003114 020100      3$:  CMP     R1,RO  ;:IS R1 SRC OK?
3449 003116 001401      BEQ     4$     ;:BRANCH IF YES
3450 003120 000000      HALT                    ;:ERROR, R1 SRC HAS SHORTED BITS
3451                                     ;:FOR LOOPING CHANGE TO "BR TST15" (602)
3452 003122 010002      4$:  MOV     RO,R2  ;:IS R2 DST OK?
3453 003124 020002      CMP     RO,R2  ;:BRANCH IF YES
3454 003126 001401      BEQ     5$     ;:ERROR, R2 DST HAS SHORTED BITS
3455 003130 000000      HALT                    ;:FOR LOOPING CHANGE TO "BR TST15" (576)
3456                                     ;:IS R2 SRC OK?
3457 003132 020200      5$:  CMP     R2,RO  ;:BRANCH IF YES
3458 003134 001401      BEQ     6$     ;:ERROR, R2 SRC HAS SHORTED BITS
3459 003136 000000      HALT                    ;:FOR LOOPING CHANGE TO "BR TST15" (573)
3460                                     ;:IS R3 DST OK?
3461 003140 010003      6$:  MOV     RO,R3  ;:BRANCH IF YES
3462 003142 020003      CMP     RO,R3  ;:ERROR, R3 DST HAS SHORTED BITS
3463 003144 001401      BEQ     7$     ;:FOR LOOPING CHANGE TO "BR TST15" (567)
3464 003146 000000      HALT                    ;:
3465

```

```

3476 003150 020300      75:  CMP      R3,R0      ; IS R3 SRC OK?
3477 003152 001401      BEQ      85          ; BRANCH IF YES
3478 003154 000000      HALT                    ; ERROR R3 SRC HAS SHORTED BITS
3479                                ; FOR LOOPING CHANGE TO "BR TST15" (564)
3480 003156 010004      85:  MOV      R0,R4
3481 003160 020004      CMP      R0,R4
3482 003162 001401      BEQ      95          ; IS R4 DST OK?
3483 003164 000000      HALT                    ; ERROR R4 DST HAS SHORTED BITS
3484                                ; FOR LOOPING CHANGE TO "BR TST15" (560)
3485 003166 020400      95:  CMP      R4,R0
3486 003170 001401      BEQ      105         ; IS R4 SRC OK?
3487 003172 000000      HALT                    ; BRANCH IF YES
3488                                ; ERROR R4 SRC HAS SHORTED BITS
3489                                ; FOR LOOPING CHANGE TO "BR TST15" (555)
3490 003174 010005      105: MOV      R0,R5
3491 003176 020005      CMP      R0,R5
3492 003200 001401      BEQ      115         ; IS R5 DST OK?
3493 003202 000000      HALT                    ; BRANCH IF YES
3494                                ; ERROR R5 DST HAS SHORTED BITS
3495                                ; FOR LOOPING CHANGE TO "BR TST15" (551)
3496 003204 020500      115: CMP      R5,R0
3497 003206 001401      BEQ      125         ; IS R5 SRC OK?
3498 003210 000000      HALT                    ; BRANCH IF YES
3499                                ; ERROR R5 SRC HAS SHORTED BITS
3500                                ; FOR LOOPING CHANGE TO "BR TST15" (546)
3501 003212 010006      125: MOV      R0,R6
3502 003214 020006      CMP      R0,R6
3503 003216 001401      BEQ      135         ; IS R6 DST OK?
3504 003220 000000      HALT                    ; BRANCH IF YES
3505                                ; ERROR R6 DST HAS SHORTED BITS
3506                                ; FOR LOOPING CHANGE TO "BR TST15" (542)
3507 003222 020600      135: CMP      R6,R0
3508 003224 001401      BEQ      145         ; IS R6 SRC OK?
3509 003226 000000      HALT                    ; BRANCH IF YES
3510                                ; ERROR R6 SRC HAS SHORTED BITS
3511                                ; FOR LOOPING CHANGE TO "BR TST15" (537)
3512 003230 005006      145: CLR      SP
3513 003232 005206      INC      SP
3514 003234 006106      ROL     SP
3515 003236 006106      ROL     SP
3516 003240 006106      ROL     SP
3517 003242 005206      INC      SP
3518 003244 006106      ROL     SP
3519 003246 006106      ROL     SP
3520 003250 006106      ROL     SP
3521 003252 006106      ROL     SP
3522 003254 006106      ROL     SP
3523 003256 006106      ROL     SP
3524 003260 005000      ;*****
3525 003262 005200      CLR      RO          ; THIS CODE
3526 003264 006100      INC      RO          ; INITIALIZES
3527 003266 006100      ROL     RO          ; THE
3528 003270 006100      ROL     RO          ; TEST
3529 003272 005200      INC      RO          ; NUMBER
3530 003274 006100      ROL     RO
3531 003276 005200      INC      RO          ; STORAGE
3532 003300 012737 000044 177770  MOV     #44,RO#177770 ; REGISTER
3533                                ; SETUP MICROPROCESSOR BREAK REG
3534                                ;*****
3535                                ; TEST 24 ONE MICROSTATE (E/CLASS*DMO*DF7)

```


3523
3524
3525
3526
3527
3528
3529
3530 003306 005200
3531 003310 005005
3532 003312 005205
3533 003314 006105
3534 003316 006105
3535 003320 005205
3536 003322 006105
3537 003324 006105
3538 003326 005205
3539 003330 006105
3540 003332
3541 003332 000264
3542 003334
3543 003334 060507
3544
3545 003336 012737 003366 000014
3546 003344 005005
3547
3548 003346 012746 000020
3549 003352 012746 003360
3550 003356 000006
3551
3552 003360 005205
3553 003362 005205
3554 003364 000240
3555 003366 012737 000016 000014
3556 003374 062706 000004
3557 003400 006005
3558 003402 103401
3559 003404 000000
3560
3561 003406
3562 003406 000000
3563
3564
3565 003410
3566 003410 001001
3567 003412 000000
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577

```

*
* THIS TEST EXECUTES AN ADD INSTRUCTION WITH SMO,DMC, AND DF7.
* IF THE TEST FAILS THE SAME FLOW (EXC. 90) IS
* TRIED WITH A PENDING BRQ (T BIT TRAP) INSTEAD OF A DF7. IF THIS TEST FAILS
* A FORK A FAILURE IS REPORTED. IF IT PASSES, A TEST 1 FAILURE IS REPORTED.
*
* ROM FLOW-30
* *****
*ST24: INC R0 ;INCREMENT TEST NUMBER
CLR R5 ;ENSURE R5 CLEAR
INC R5 ;SET R5 EQUAL
ROL R5 ;TO 52
ROL R5 ;WHICH
INC R5 ;WILL CAUSE
ROL R5 ;A JUMP
ROL R5 ;TO TESTCC
INC R5
ROL R5 ;TEST
SYNC24: SEZ ;ENSURE Z SET
IUT24: ADD R5,PC ;ADD SHOULD SKIP TO TAG TESTCC
;FAILURE- TRY E/CLASS*BRQ*DMO
MOV #FORKA,2#14 ;SETUP VECTOR FOR RETURN
CLR R5 ;ENSURE R5 CLEAR
;*****
MOV #BIT4, -(SP) ;THIS CODE
MOV #15, -(SP) ;SETS THE
RTT ;T BIT
;*****
IS: INC R5 ;TRAP HERE IF FORK A OK
INC R5 ;WILL EXECUTE IF FORK A FAILED
NOP ;ALLOW T BIT TRAP IF FORK FAILED
FORKA: MOV #16,2#14 ;RESTORE T BIT VECTOR
ADD #4,SP ;RESTORE SP
ROR R5 ;IS R5 1 OR 2?
BCS IS ;BRANCH ON 1 (FORK A OK)
HALT ;FORK A FAILURE INTO ROM STATE EXC.90
;FOR LOOPING CHANGE TO "BR TST24+2" (741)
IS: HALT ;EITHER PCB DID NOT LOAD OR RACH DF7 STUCK HIGH
;FOR LOOPING CHANGE TO "BR TST24+2" (740)
TESTCC: BNE TST25 ;GO TO NEXT TEST IF CCLDS OK
HALT ;STATE EXC.90 BAD
;FOR LOOPING CHANGE TO "BR TST24+2" (736)
.SBTTL
;*****
;TEST 25 TWO MICROSTATES (NEG*DMO)
*
* THIS TEST EXECUTES A NEGATE INSTRUCTION WITH DMO.
*
* IF FORK A FAILS EXECUTION WOULD GO TO EITHER RSD.00, ZAP.00,
* OR FOP.00.
* FOP.00 WILL CAUSE THE PROCESSOR TO HANG.
*
```

```

3578      *      R5D.00 WOULD CAUSE A TRAP TO LOCATION 10. THIS WOULD ONLY HAPPEN
3579      *      IF RACH NEG.B*DMO DID NOT GO HIGH.
3580      *      ZAP.00 WOULD CAUSE A TRAP TO LOCATION 24. THIS WOULD ONLY HAPPEN
3581      *      IF RACH A2 RAB00 DID NOT GO LOW.
3582      *
3583      *
3584      *      ROM FLOW-301,210
3585      *      *****
3585 003414 005200 TST25: INC R0 ;INCREMENT TEST NUMBER
3586 003416 005004 CLR R4 ;INITIALIZE CC ERROR RECORD
3587 003420 005005 CLR R5 ;SET UP R5
3588 003422 005205 INC R5 ;FOR TEST
3589 003424 000257 CCC ;
3590 003426 SYNC25:
3591 003426 000266 +SEZ!SEV ;CC'S=0110
3592 003430 IUT25:
3593 003430 005405 NEG R5 ;EXECUTE NEGATE CC'S=1001
3594 003432 000401 BR 1$ ;GET OVER ERROR CALL
3595 003434 000000 HALT ;RACH E57(6) DOES NOT GO LOW
3596 ;FOR LOOPING CHANGE TO "BR TST25" (767)
3597 003436 100003 1$: BPL NEGR5
3598 003440 001402 BEQ NEGR5
3599 003442 102401 BVS NEGR5
3600 003444 103401 BCS +4
3601 003446 005204 NEGR5: INC R4 ;ERROR, INCORRECT CC'S AFTER NEG.
3602 003450 005001 CLR R1 ;SET R1 TO
3603 003452 005301 DEC R1 ;-1 WITHOUT NEGATING
3604 003454 020501 CMP R5,R1 ;DID R5 GET -1?
3605 003456 001414 BEQ R5OK ;BRANCH IF R5 OK
3606 003460 005704 TST R4 ;IS THERE A CC PROBLEM?
3607 003462 001405 BEQ 3$ ;BRANCH IF NO
3608 003464 022705 177776 CMP #177776,R5 ;DID NEG ONE'S COMPLEMENT?
3609 003470 001001 BNE 2$ ;BRANCH IF NO
3610 003472 000000 HALT ;CC'S BAD AND NEG.90 DID NOT ADD 1
3611 ;FOR LOOPING CHANGE TO "BR TST25+2" (751)
3612 003474 2$: HALT ;CC'S BAD AND R5 BAD
3613 003474 000000 ;FOR LOOPING CHANGE TO "BR TST25+2" (750)
3614 ;DID NEGATE DO A ONE'S COMPLIMENT?
3615 003476 022705 177776 3$: CMP #177776,R5 ;BRANCH IF NO
3616 003502 001001 BNE 4$ ;CC'S OK BUT NEG.90 DID NOT ADD 1
3617 003504 000000 HALT ;FOR LOOPING CHANGE TO "BR TST25+2" (744)
3618 ;CC'S OK BUT R5 BAD
3619 003506 4$: HALT ;FOR LOOPING CHANGE TO "BR TST25+2" (743)
3620 003506 000000 ;CC PROBLEM?
3621 ;GO TO NEXT TEST IF NO
3622 003510 005704 R5OK: TST R4 ;NEGATE OK BUT INCORRECT CC'S
3623 003512 001401 BEQ TST26 ;FOR LOOPING CHANGE TO "BR TST25+2" (740)
3624 003514 000000 HALT
3625
3626 .SBTTL
3627 *****
3628 *TEST 26 THREE MICROSTATES (BIN*SM1*DMO*-DF7*SRO(0))
3629 *
3630 *
3631 * IF FORK A FAILS EXECUTION WILL GO TO EITHER EXEC.80 OR D12.00.
3632 * EXC.80 WILL HANG THE PROCESSOR IN THE PAUSE STATE AT MICRO ADDRESS 343.
3633 * THIS WILL ONLY HAPPEN IF RACH RAD00 IS NOT GOING LOW DUE
* TO RACH R1 RAB00 (AFIR59(1))*[-BIN+SMO1]*U/CLASS).

```

```

3634 : * D12.00 WOULD MOV THE PC TO LOCATION D.
3635 : *
3636 : * IF FORK C FAILS EXECUTION WOULD GO FROM S13.10 TO D00.80 OR
3637 : * D45.01 OR S13.20 OR D12.00 OR JSR.10 OR ASC.80 OR RTI.50 OR ASH.20 OR FCP.50.
3638 : * D00.80 WOULD SWAP THE BYTES OF THE SOURCE OPERAND BEFORE
3639 : * PUTTING THEM IN R5.
3640 : * D45.01 WOULD MOVE THE PC TO LOCATION D.
3641 : * S13.20 WILL EXECUTE A SM3 (AND NO AUTO INC) INSTR. THIS WILL CAUSE
3642 : * AN ODD ADDRESS TRAP SINCE LOCATION POSERR CONTAINS AN ODD WORD.
3643 : * JSR.10 WOULD PUSH THE ADDR OF POSERR ONTO THE STACK.
3644 : * ASC.80 WILL HALT AT 8$.
3645 : * RTI.50 WILL CAUSE 1004XX TO BE PLACED IN THE PS WORD
3646 : * AND THE PROCESSOR WILL TRAP TO LOCATION 14.
3647 : * FOP.50 WILL "????".
3648 : * ASH.20 WOULD CAUSE A BAD CC.
3649 : * IF THE SRC CONST FAILS IN STATE S13.00 AND ADDS 1 OR 3, AN ODD
3650 : * ADDRESS TRAP WILL OCCUR.
3651 : * IF THE SRC CONSTANT ADDS 2, THE ERROR AT 6$ WILL REPORT THE FAILURE.
3652 : *
3653 : *
3654 : * ROM FLOW-21,27,205

```

```

*****
3655 003516 005200 TST26: INC R0 ;INCREMENT TEST NUMBER
3656 003520 005005 CLR R5 ;SETUP R5
3657 003522 SYNC26:
3658 003522 000244 CLZ ;ENSURE Z CLEAR TO CATCH A FAILURE TO ASC.80
3659 003524 IUT26:
3660 003524 011705 MOV (PC), R5 ;MOVE 1004XX TO R5 (XX MUST BE
3661 ;ODD TO CATCH A FAILURE TO S13.20)
3662 003526 100443 POSERR: BMI 7$ ;BRANCH IF CC OK (ENSURE OFFSET IS ODD)
3663 003530 100441 BMI 6$ ;BRANCH IF SRC CONST ADDED 2
3664 ;FAILURE
3665 003532 013767 177776 175436 MOV 2#PSW, 2#PSW ;SAVE ERROR PSW
3666 003540 022737 003526 000000 1$: CMP #POSERR, 2#0 ;DID FORK A GO TO D12.00 OR FORK C TO D45.01?
3667 003546 001006 BNE 3$ ;BRANCH IF NO
3668 ;EITHER FORK A OR C FAILED-FIND OUT WHICH ONE
3669 003550 005005 CLR R5 ;SETUP R5
3670 003552 012501 MOV (R5)+, R1 ;TEST TO SEE IF FORK A OR FORK C FAILED
3671 003554 005705 TST R5 ;DID R5 INCREMENT
3672 003556 001401 BEQ 2$ ;BRANCH IF NO
3673 003560 000000 HALT ;FORK C FAILED, EITHER IRCC DMD NOT GETTING
3674 ;THRU TO RACL RADRO7 OR IRCC DMD IS STUCK HIGH
3675 ;FOR LOOPING CHANGE TO "BR TST26+2" (757)
3676 003562 2$:
3677 003562 000000 HALT ;FORK A FAILED, RACH A1 RAB04 NOT GOING LOW
3678 ;DUE TO EITHER RACE: BF1=7 OR
3679 ;BF1=0 OR SMO STUCK LOW OR RACH E11 BAD
3680 ;FOR LOOPING CHANGE TO "BR TST26+2" (756)
3681 003564 000305 3$: SWAB R5 ;SETUP R5 TO TEST IF INSTR. WENT THRU D00.80
3682 003566 020537 003526 CMP R5, 2#POSERR ;DID INSTR GO THRU D00.3?
3683 003572 001001 BNE 4$ ;BRANCH IF NO
3684 003574 000000 HALT ;IRCC RAB00 NOT GETTING TO RACL RADRO0
3685 ;FOR LOOPING CHANGE TO "BR TST26+2" (751)
3686 003576 026627 000010 003526 4$: CMP 10(SP), #POSERR ;DID FORK C GO TO JSR.10?
3687 003604 001001 BNE 5$ ;BRANCH IF NO
3688 003606 000000 HALT ;INPUT TO IRCC E40 PIN 5 STUCK HIGH
3689 ;FOR LOOPING CHANGE TO "BR TST26+2" (744)

```

F07

```

3690 003610 032767 000017 175360 5$: BIT #17,SERPSW ;DID ALL CC'S CLEAR?
3691 003616 001404 BEQ 9$ ;BRANCH IF YES
3692 003620 032767 000020 175350 BIT #BIT4,SERPSW ;DID Z BIT SET?
3693 003626 001401 BEQ 8$ ;BRANCH IF NO
3694 003630 9$: HALT ;BAD CONDITION CODE
3695 003630 000000 ;FOR LOOPING CHANGE TO "BR TST26+2" (733)
3696
3697 003632 8$: HALT ;EITHER INPUT TO C MUX STUCK LOW OR MUX BAD OR
3698 003632 000000 ;CO RAB01 STUCK LOW OR RACL RADR01 INPUT STUCK LOW
3699 ;FOR LOOPING CHANGE TO "BR TST26+2" (732)
3700
3701 003634 6$: HALT ;SRC CONST EQUAL TO 2 SHOULD BE 0
3702 003634 000000 ;FOR LOOPING CHANGE TO "BR TST26+2" (731)
3703
3704 003636 000241 7$: CLC ;ENSURE C CLEAR
3705 003640 006105 ROL R5 ;CHECK IF R5 WAS LOADED
3706 003642 103401 BCS TST27 ;GO TO NEXT TEST IF C SET
3707 003644 000000 HALT ;ERROR, R5 DID NOT LOAD
3708 ;FOR LOOPING CHANGE TO "BR TST26+2" (725)
3709
3710 ;*****
3711 ;TEST 27 THREE MICROSTATES (BIN*SM2*DMO*-DF7*SRO(0))
3712 ;
3713 ; THIS TEST IS THE SAME AS THE PREVIOUS TEST EXCEPT FOR THE
3714 ; SM. A WORD AND BYTE INSTRUCTION IS EXECUTED TO VERIFY THAT STATE
3715 ; S13.01 ADDS THE CORRECT SOURCE CONSTANT.
3716 ;
3717 ; IF FORK A FAILS EXECUTION WILL GO TO EXC.80.
3718 ; EXC.80 WILL HANG THE PROCESSOR IN THE PAUSE STATE AT MICRO ADDRESS 343.
3719 ; THIS WILL ONLY HAPPEN IF RACL RADR01
3720 ; IS NOT GOING LOW DUE TO RACF A1 RAB01 (AFIR10(1)*U/CLASS).
3721 ;
3722 ; ROM FLOW-22,27,205
3723 003646 005200 TST27: INC R0 ;INCREMENT TEST NUMBER
3724 003650 005005 CLR R5 ;SETUP R5
3725 003652 000240 SYNC27: NOP
3726 003654 IUT27:
3727 003654 012705 MOV (PC)+,R5 ;MOVE 000401 TO R5
3728 003656 000401 ER25: BR 3$
3729 003660 000412 BR 4$
3730 ;FAILURE-SOURCE CONSTANT FAILED. TRY SM3
3731 003662 012705 011704 3$: MOV #SUBTAB,R5 ;GET ADDRESS OF LOCATION THAT CONTAINS ADDR.
3732 003666 010501 MOV R5,R1 ;SAVE R5 IN R1
3733 003670 013502 MOV 2(R5)+,R2 ;EXECUTE AN SM3 INSTRUCTION
3734 003672 005201 INC R1 ;ADJUST R1 TO
3735 003674 005201 INC R1 ;LOOK LIKE R5
3736 003676 020501 CMP R5,R1 ;DID R5 AUTO INCREMENT?
3737 003700 001401 BEQ 2$ ;BRANCH IF YES
3738 003702 000000 HALT ;SOURCE CONST FAILURE ON IRC
3739 ;FOR LOOPING CHANGE TO "BR TST27+2" (762)
3740
3741 003704 2$: HALT ;SRC CONST FAILURE EITHER ON IRC OR DAP
3742 ;FOR LOOPING CHANGE TO "BR TST27+2" (761)
3743
3744 003706 010705 4$: MOV PC,R5 ;SETUP R5 TO HOLD ADDRESS
3745 003710 010501 MOV R5,R1 ;SAVE R5 IN R1
3745 003712 !12502 MOVB (R5)+,R2 ;TEST TO SEE IF SCR CONST=1 ON BYTE
  
```

```

3746 003714 005201      INC      R1      ;SETUP R1 TO LOOK LIKE R5
3747 003716 020501      CMP      R5,R1   ;DID R5 AUTOINCREMENT BY 1?
3748 003720 001410      BEQ      TST30   ;BRANCH IF YES
3749                ;FAILURE-SOURCE CONSTANT FAILED ON BYTE. TRY SM4
3750 003722 010705      MOV      PC,R5   ;PUT ADDRESS IN R5
3751 003724 010501      MOV      R5,R1   ;SAVE R5 IN R1
3752 003726 114502      MOVB    -(R5),R2 ;EXECUTE AN SM4 INSTRUCTION
3753 003730 005301      DEC      R1      ;ADJUST R1 TO LOOK LIKE R5
3754 003732 020501      CMP      R5,R1   ;DID R5 AUTO DECREMENT?
3755 003734 001001      BNE     5$       ;BRANCH IF NO
3756 003736 000000      HALT                    ;IRCC SRCM2 STUCK HIGH INTO IRC E8
3757                ;FOR LOOPING CHANGE TO "BR TST27+2" (744)
3758 003740          5$:          HALT                    ;BYTE SRC CONST FAILURE EITHER ON IRC OR DAP
3759 003740 000000          ;FOR LOOPING CHANGE TO "BR TST27+2" (743)
3760                ;*****
3761                ;*TEST 30      ALU CARRY FUNCTIONAL TEST
3762                ;*
3763                ;*
3764                ;* THIS TEST DOES A COMPLETE CHECK OF THE ALU CARRY FUNCTIONS
3765                ;* THE FIRST SECTION ENSURES THAT ALL THE INPUT AND OUTPUT LINES ARE
3766                ;* OK AND THE REST OF THE TEST ENSURES THAT THE CARRY LOGIC IS OK.
3767                ;* FOLLOWING ARE THE LOGIC EQUATIONS FOR A 745181 AND 745182 THAT
3768                ;* DICTATED THE PATTERNS USED IN EACH SECTION:
3769                ;*
3770                ;* 745181
3771                ;* G=A3*B3+A2*B2*(A3+B3)+A1*B1*(A2+B2)*(A3+B3)+A0*B0*(A1+B1)*(A2+B2)*(A3+B3
3772                ;* P=(A3+B3)*(A2+B2)*(A1+B1)*(A0+B0)
3773                ;* COUT=G+P*CIN
3774                ;* 745182
3775                ;* CX=G0+P0*CIN
3776                ;* CY=G1+P1*G0+P1*P0*CIN
3777                ;* CZ=G2+P2*G1+P2*P1*G0+P2*P1*P0*CIN
3778 003742 005200      TST30: INC      R0      ;*****
3779                ;SECTION 1-INPUT/OUTPUT BIT TEST
3780 003744 012701 125252      MOV      #125252,R1 ;PUT DATA PATTERN IN R1
3781 003750 062701 052525      ADD      #52525,R1  ;ADD COMPLIMENT PATTERN
3782 003754 005101          COM      R1          ;MAKE RESULT 0
3783 003756 001401          BEQ      2$         ;BRANCH IF IT IS ZERO
3784 003760 000000          HALT                    ;BIT FAILED DURING ADD
3785                ;FOR LOOPING CHANGE TO "BR TST30+2" (771)
3786 003762 012701 052525      2$:      MOV      #52525,R1 ;PUT PATTERN IN R1
3787 003766 062701 125252      ADD      #125252,R1 ;ADD COMPLIMENT PATTERN
3788 003772 005101          COM      R1          ;MAKE IT ZERO
3789 003774 001401          BEQ      3$         ;BRANCH IF IT WENT TO ZERO
3790 003776 000000          HALT                    ;BIT FAILED DURING ADD
3791                ;FOR LOOPING CHANGE TO "BR 2$" (771)
3792                ;*****
3793                ;SECTION 2-G=A3*B3
3794 004000 012702 167357      3$:      MOV      #167357,R2 ;PUT COMPLIMENT OF EXPECTED PATTERN IN R2
3795 004004 012701 104210      MOV      #104210,R1 ;PUT PATTERN IN R1
3796 004010 060101          ADD      R1,R1       ;ADD IT TO ITSELF
3797 004012 103401          BCS     4$         ;BRANCH IF DAPH COUT15 OK
3798 004014 000000          HALT                    ;DAPH COUT15 DID NOT GO LOW
3799                ;FOR LOOPING CHANGE TO "BR 3$" (771)
3800 004016 050201      4$:      BIS      R2,R1     ;MAKE R1 -1
3801 004020 005101          COM      R1          ;MAKE IT ZERO
    
```

```

3802 004022 001401      BEQ 55      ;BRANCH IF IT IS
3803 004024 000000      HALT      ;A G LINE FAILED
3804                                     ;FOR LOOPING CHANGE TO "BR 35" (765)
3805 ;*****
3806 ;SECTION 3-G=A2*B2*(A3+B3)
3807 004026 012701 146314 55:  MOV  #146314,R1  ;PUT PATTERN IN R1
3808 004032 062701 042104  ADD  #42104,R1   ;ADD PATTERN
3809 004036 050201      BIS  R2,R1      ;MAKE R1 -1
3810 004040 005101      COM  R1        ;MAKE IT ZERO
3811 004042 001401      BEQ 65      ;BRANCH IF IT IS ZERO
3812 004044 000000      HALT      ;A G LINE FAILED
3813                                     ;FOR LOOPING CHANGE TO "BR 55" (770)
3814 004046 012701 042104 65:  MOV  #42104,R1  ;REVERSE INPUTS
3815 004052 062701 146314  ADD  #146314,R1 ;TC ALU
3816 004056 050201      BIS  R2,R1      ;MAKE R1 -1
3817 004060 005101      COM  R1        ;MAKE IT ZERO
3818 004062 001401      BEQ 75      ;BRANCH IF IT IS
3819 004064 000000      HALT      ;A G LINE FAILED
3820                                     ;FOR LOOPING CHANGE TO "BR 65" (770)
3821 ;*****
3822 ;SECTION 4-G=A1*B1*(A2+B2)*(A3+B3)
3823 004066 012701 167356 75:  MOV  #167356,R1 ;PUT PATTERN IN R1
3824 004072 062701 021042  ADD  #21042,R1  ;ADD PATTERN
3825 004076 050201      BIS  R2,R1      ;MAKE R1 -1
3826 004100 005101      COM  R1        ;MAKE IT ZERO
3827 004102 001401      BEQ 85      ;BRANCH IF IT IS
3828 004104 000000      HALT      ;A G LINE FAILED
3829                                     ;FOR LOOPING CHANGE TO "BR 75" (770)
3830 004106 012701 021042 85:  MOV  #21042,R1  ;REVERSE INPUTS
3831 004112 062701 167356  ADD  #167356,R1 ;TO THE ALU
3832 004116 050201      BIS  R2,R1      ;MAKE R1 -1
3833 004120 005101      COM  R1        ;MAKE IT ZERO
3834 004122 001401      BEQ 95      ;BRANCH IF IT IS
3835 004124 000000      HALT      ;A G LINE FAILED
3836                                     ;FOR LOOPING CHANGE TO "BR 85" (770)
3837 ;*****
3838 ;SECTION 5-G=A0*B0*(A1+B1)*(A2+B2)*(A3+B3)
3839 004126 012701 177777 95:  MOV  #-1,R1    ;PUT PATTERN IN R1
3840 004132 062701 010421  ADD  #10421,R1  ;ADD PATTERN TO R1
3841 004136 050201      BIS  R2,R1      ;MAKE R1 -1
3842 004140 005101      COM  R1        ;MAKE IT ZERO
3843 004142 001401      BEQ 105     ;BRANCH IF IT IS
3844 004144 000000      HALT      ;A G LINE FAILED
3845                                     ;FOR LOOPING CHANGE TO "BR 95" (770)
3846 004146 012701 010421 105: MOV  #10421,R1  ;REVERSE INPUTS
3847 004152 062701 177777  ADD  #-1,R1    ;TO THE ALU
3848 004156 050201      BIS  R2,R1      ;MAKE R1 -1
3849 004160 005101      COM  R1        ;MAKE IT ZERO
3850 004162 001401      BEQ 115     ;BRANCH IF IT IS
3851 004164 000000      HALT      ;A G LINE FAILED
3852                                     ;FOR LOOPING CHANGE TO "BR 105" (770)
3853 ;*****
3854 ;SECTION 6-P OUTPUTS AND CX=P0*CIN, CY=P1*P0*CIN, CZ=P2*P1*P0*CIN
3855 004166 012701 177777 115: MOV  #-1,R1   ;PUT PATTERN IN R1
3856 004172 000261      SEC          ;SET C
3857 004174 005501      ADC  R1      ;CAUSES CARRY TO GO ALL THE WAY

```

3858	004176	103401		BCS	12\$; BRANCH IF CARRY CAME OUT
3859	004200	000000		HALT		; DAPH COUT15 DID NOT GO LOW
3860						; FOR LOOPING CHANGE TO "BR 11\$" (772)
3861	004202	001401		12\$: BEQ	13\$; BRANCH IF R1 WENT TO ZERO
3862	004204	000000		HALT		; EITHER A P LINE OR THE 745182 FAILED
3863						; FOR LOOPING CHANGE TO "BR 11\$" (770)

3865				SECTION 7-CY=P1*GO		
3866	004204	012701	000370	13\$: MOV	#370,R1	; PUT PATTERN IN R1
3867	004206	062701	000010	ADD	#10,R1	; ADD DATA TO R1
3868	004216	052701	177377	BIS	#177377,R1	; MAKE R1 -1
3869	004222	005101		COM	R1	; MAKE IT ZERO
3870	004224	001401		BEQ	14\$; BRANCH IF IT WORKED
3871	004226	000000		HALT		; DAPF E44 FAILED
3872						; FOR LOOPING CHANGE TO "BR 13\$" (767)

3875	004230	012701	007600	SECTION 8-CZ=P2*G1		
3876	004234	062701	000200	14\$: MOV	#7600,R1	; PUT DATA IN R1
3877	004240	052701	167777	ADD	#200,R1	; ADD DATA TO R1
3878	004244	005101		BIS	#167777,R1	; MAKE R1 -1
3879	004246	001401		COM	R1	; MAKE IT ZERO
3880	004250	000000		BEQ	15\$; BRANCH IF IT WORKED
3881				HALT		; DAPF E44 FAILED
3882						; FOR LOOPING CHANGE TO "BR 14\$" (767)

3884	004252	012701	007770	SECTION 9-CZ=P2*P1*GO		
3885	004256	062701	000010	15\$: MOV	#7770,R1	; PUT DATA IN R1
3886	004262	052701	167777	ADD	#10,R1	; ADD DATA TO R1
3887	004266	005101		BIS	#167777,R1	; MAKE R1 -1
3888	004270	001401		COM	R1	; MAKE IT ZERO
3889	004272	000000		BEQ	TST31	; BRANCH IF IT WORKED
3890				HALT		; DAPF E44 FAILED
3891						; FOR LOOPING CHANGE TO "BR 15\$" (767)

TEST 31 THREE MICROSTATES (DAC*DM2*0/CLASS)

*
* IF FORK A FAILS EXECUTION WILL GO TO RSD.00. THIS WILL ONLY HAPPEN
* IF RACE AD RAB01 DOES NOT GO LOW OR DOES NOT GET THRU
* TO RACL RAD01. THIS WILL CAUSE A TRAP TO LOCATION 10.
*
* IF BEN15 FAILS EXECUTION WILL GO TO D45.80.
* THIS WILL CAUSE A TRAP TO 4 WITH THE ADDRESS OF 1\$ ON THE STACK.
*
* IF THE DESTINATION CONSTANT FAILS(ADDS 1 OR 3) IN STATE D12.60
* AN ODD ADDRESS TRAP WILL OCCUR.
* IF THE DST CONST ADDS 0, THE ERROR AT 3\$ WILL REPORT THE FAILURE.
* IF THE DESTINATION IS NOT LOADED WITH THE SOURCE, THE
* ERROR AFTER 5\$ WILL REPORT THE FAILURE.
*
* ROM FLOW-2,155,312

3909	004274	005200		TST31: INC	RO	; INCREMENT TEST NUMBER
3910	004276	012705		MOV	(PC)+,R5	; PUT "BR 4\$" IN R5
3911	004300	000401		.WORD	000401	; CONTAINS BINARY OF "BR 4\$"
3912	004302	000240		SYNC31: NOP		
3913	004304			IUT:		

```

3914 004304 010527          1$:  MOV      R5,(PC)+      ;EXECUTE INSTRUCTION UNDER TEST
3915 004306 000401          BR      4$              ;WILL EXECUTE THIS IF INSTR FAILS TO AUTO INC
3916 004310 000415          BR      5$              ;AUTO INC OK, GO CHECK IF LOAD OK
3917                                     ;FAILURE-DESTINATION CONSTANT FAILED. TRY DM4.
3918 004312 012705 001164  4$:  MOV      #STMP1,R5      ;PUT ADDRESS IN R5
3919 004316 010125          MOV      R1,(R5)+      ;EXECUTE INSTRUCTION UNDER TEST
3920 004320 022705 001164  CMP      #STMP1,R5      ;DID R5 STAY THE SAME?
3921 004324 001006          BNE     2$              ;BRANCH IF NO
3922 004326 010145          MOV      R1, -(R5)      ;SEE IF AUTO DEC. WORKS
3923 004330 022705 001162  CMP      #STMP1-2,R5    ;DID R5 AUTO DEC?
3924 004334 001001          BNE     3$              ;BRANCH IF NO
3925 004336 000000          HALT                    ;AUTO DEC WORKS SO ROM STATE D12.60 PROBABLY BAD
3926                                     ;FOR LOOPING CHANGE TO "BR TST31+2" (757)
3927 004340          3$:  HALT                    ;IRCD DSTCON=2 EITHER STUCK LOW OR
3928 004340 000000          ;NOT GETTING THRU KOMUX
3929                                     ;FOR LOOPING CHANGE TO "BR TST31+2" (756)
3930
3931 004342          2$:  HALT                    ;BEN15 OK & DST CONST OK BUT INSTR DOSEN'T WORK
3932 004342 000000          ;FOR LOOPING CHANGE TO "BR TST31+2" (755)
3933
3934 004344 012705 004304  5$:  MOV      #1$,R5        ;GET ADDR OF INSTR UNDER TEST
3935 004350 011505          MOV      (R5),R5        ;GET INSTR UNDER TEST
3936 004352 022705 000401  CMP      #401,R5        ;DID AUTO DEC OCCUR?
3937 004356 001005          BNE     8$              ;BRANCH IF NO
3938 004360 012705 010027  MOV      #10027,R5      ;GET OP CODE OF INSTR UNDER TEST
3939 004364 010567 177714  MOV      R5,1$          ;RESTORE INSTR UNDER TEST
3940 004370 000000          HALT                    ;EITHER RACK BRCA05 NOT GOING LOW OR
3941                                     ;IT IS NOT GETTING THRU RA0L RAD05
3942                                     ;FOR LOOPING CHANGE TO "BR TST31+2" (742)
3943 004372 010527          8$:  MOV      R5,(PC)+      ;TEST TO SEE IF (PC)+ WAS LOADED
3944 004374 100000          7$:  .WORD    100000      ;STORAGE LOCATION FOR PREVIOUS INSTR.
3945 004376 012701 004374  MOV      #7$,R1         ;PUT ADDRESS OF 7$ IN R1
3946 004402 011102          MOV      (R1),R2        ;GET CONTENTS OF 7$
3947 004404 100001          BPL     6$              ;BRANCH IF LOAD OK
3948 004406 000000          HALT                    ;ERROR, (PC)+ DID NOT LOAD CORRECTLY
3949                                     ;FOR LOOPING CHANGE TO "BR TST31+2" (733)
3950 004410 012702 100000  6$:  MOV      #BIT15,R2    ;SET SIGN BIT IN R2
3951 004414 010221          MOV      R2,(R1)+      ;RESTORE 7$
3952                                     ;*****
3953                                     ;*TEST 32      THREE MICROSTATES (DAC*DM1*0/CLASS)
3954                                     ;*
3955                                     ;* THIS TEST IS THE SAME AS THE PREVIOUS TEST EXCEPT FOR THE DM.
3956                                     ;* IF FORK A FAILS, EXECUTION WILL GO TO RSD.00.
3957                                     ;* THIS WILL CAUSE A TRAP TO LOCATION 10 WITH AN ODD ADDRESS ERROR.
3958                                     ;* THIS WILL ONLY HAPPEN IF RACE AD RAB00 DOES NOT GO LOW OR
3959                                     ;* DOES NOT GET TO RA0L.
3960                                     ;* EITHER E44 OR E6 IS BAD.
3961                                     ;*
3962                                     ;* IF THE INSTRUCTION FAILS TO MOVE R5 TO THE PC INDIRECT,
3963                                     ;* THE ERROR AFTER 1$ WILL REPORT THE FAILURE.
3964                                     ;*
3965                                     ;* ROM FLOW-1,155,312
3966                                     ;*****
3967 004416 005200  †ST32: INC      R0          ;INCREMENT TEST NUMBER
3968 004420 012705          MOV      (PC)+,R5      ;PUT BR IN R5
3969 004422 000401          .WORD    000401        ;BINARY WORD FOR BR .+2
    
```


K07

3970	004424	000240		SYNC32: NOP	
3971	004426			IUT32:	
3972	004426	010517		15: MOV R5, (PC)	; EXECUTE INSTRUCTION UNDER TST
3973	004430	000240		.WORD 240	; SHOULD REPLACE THIS WITH BR 25
3974	004432	000000		HALT	; LOCATION POINTED TO BY PC DID NOT LOAD
3975					; FOR LOOPING CHANGE TO "BR TST32+2" (772)
3976	004434	012705	000240	25: MOV #240, R5	; THIS CODE
3977	004440	012701	004430	MOV #15, R1	; RESTORES THE NOP
3978	004444	010511		MOV R5, (R1)	; AT LOCATION 15
3979	004446	000264		SEZ	; ENSURE Z SET
3980	004450	010517		MOV R5, (PC)	; EXECUTE INSTRUCTION UNDER TEST
3981	004452	000000		.WORD 0	
3982	004454	001001		BNE TST33	; CC OK, GO TO NEXT TEST
3983	004456	000000		HALT	; STATE D12.20 BAD
3984					; FOR LOOPING CHANGE TO "BR TST32+2" (760)

 TEST 33 THREE MICROSTATES (DAC*DM4*0/CLASS)

IF FORK A FAILS EXECUTION WILL GO TO RSD.00.
 THIS WILL CAUSE A TRAP TO LOCATION 10. THIS WILL ONLY HAPPEN IF
 RACE AD RAB02 IS NOT GOING LOW. EITHER RACE E33 OR
 E6(988) IS BAD.

IF THE DST CONST FAILS TO SUBTRACT 2, THE ERROR AT EITHER 15
 OR 15-2 WILL REPORT THE FAILURE.

IF BENO1 FAILS (CAUSED BY IRCD DM357 STUCK HIGH)
 EXECUTION WILL GO TO D10.00 WHICH WILL EXECUTE A MODE 5
 INSTEAD OF MODE 4.

IF THE DESTINATION IS NOT LOADED PROPERLY,
 THE ERROR AT 35 WILL REPORT THE FAILURE.

ROM FLOW-4, 122, 157

 TST33: INC R0 ; INCREMENT TEST NUMBER
 MOV #1, R2 ; SETUP R2
 MOV #STMP2, R5 ; PUT ADDRESS OF STMP2 IN R5
 MOV R5, R1 ; SAVE R5

 ; THESE THREE INSTRUCTIONS ARE USED TO CATCH IRCD DM357 STUCK HIGH
 MOV #STMP1, R3 ; PUT ADDR OF STMP1 IN R3
 MOV R5, (R3) ; PUT ADDR. OF STMP2 IN STMP1

 SYNC33: NOP
 IUT33:

MOV R2, -(R5)	; EXECUTE INSTRUCTION UNDER TEST
CMP R5, R3	; DID R5 AUTO DECREMENT?
BEQ 45	; BRANCH IF YES.
MOV #STMP1+2, R5	; PUT ADDR. IN R5
MOV #STMP1, R1	; PUT ADDR IN R1
MOV (R1), -(R5)	; EXECUTE INSTRUCTION
CMP R1, R5	; DID R5 AUTO DECREMENT?
BEQ 15	; BRANCH IF YES
HALT	; IRCC DSTMP4 STUCK HIGH
	; FOR LOOPING CHANGE TO "BR TST33+2" (754)

4000			
4001			
4002			
4003			
4004			
4005	004460	005200	
4006	004462	012702	000001
4007	004466	012705	001166
4008	004472	010501	
4009			
4010			
4011	004474	012703	001164
4012	004500	010513	
4013			
4014	004502	000240	
4015	004504		
4016	004504	010245	
4017	004506	020503	
4018	004510	001411	
4019	004512	012705	001166
4020	004516	012701	001164
4021	004522	011145	
4022	004524	020105	
4023	004526	001401	
4024	004530	000000	
4025			

```

4026 004532          1$:
4027 004532 000000          HALT          ;IRCC DSTM4 OK BUT D45.00 DOES NOT DECREMENT
4028                                     ;FOR LOOPING CHANGE TO "BR TST33+2" (753)
4029 004534 011505          4$:  MOV      (R5),R5          ;GET CONTENTS OF $TMP1
4030 004536 020205          CMP      R2,R5          ;DID DEST. GET LOADED?
4031 004540 001005          BNE     2$             ;BRANCH IF NO
4032 004542 005205          INC     R5             ;ADJUST R5 TO EVEN ADDRESS
4033 004544 000264          SEZ     Z             ;ENSURE Z SET
4034 004546 010245          MOV     R2, -(R5)      ;EXECUTE INSTRUCTION UNDER TEST
4035 004550 001020          BNE     TST34         ;CC'S OK
4036 004552 000000          HALT          ;STATE D10.40 BAD
4037                                     ;FOR LOOPING CHANGE TO "BR TST33+2" (743)
4038 004554 011101          2$:  MOV     (R1),R1      ;GET CONTENTS OF $TMP2
4039 004556 020201          CMP     R2,R1        ;DID A MODE 5 TAKE PLACE?
4040 004560 001001          BNE     3$           ;BRANCH IF NO
4041 004562 000000          HALT          ;EITHER IRCD DM357 STUCK HIGH OR RACK E49(C1) BAD
4042                                     ;FOR LOOPING CHANGE TO "BR TST33+2" (737)
4043 004564          3$:
4044 004564 000000          HALT          ;DST($TMP1) DID NOT GET LOADED FROM SRC(R2)
4045                                     ;FOR LOOPING CHANGE TO "BR TST33+2" (736)
4046                                     ;*****
4047 :*TEST 34          THREE MICROSTATES (DAC*DM1*TST.B*DRO(0))
4048 :*
4049 :*
4050 :*          IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TRAP TO 10.
4051 :*          THIS WILL ONLY HAPPEN IF RA SAME?
4052 004566 001372          BNE     2$           ;BRANCH IF NO
4053 004570 010145          MOV     R1, -(R5)    ;SEE IF AUTO DEC. WORKS
4054 004572 022705 001162  CMP     #TMP1-2,R5   ;DID R5 AUTO DEC?
4055 004576 001372          BNE     3$           ;BRANCH IF NO
4056 004600 000000          HALT          ;AUTO DEC WORKS SO ROM STATE D12.60 PROBABLY BAD
4057 004602          3$:
4058 004602 000000          HALT          ;FOR LOOPING CHANGE TO "BR TST33+2" (730)
4059                                     ;IRCC DSTCON=2 EITHER STUCK LOW OR
4060                                     ;NOT GETTING THRU KOMUX
4061 004604          2$:
4062 004604 000000          HALT          ;FOR LOOPING CHANGE TO "BR TST33+2" (727)
4063                                     ;BEN15 OK & DST CONST OK BUT INSTR DOSEN'T WORK
4064 004606 012705 004532  5$:  MOV     #1$,R5       ;FOR LOOPING CHANGE TO "BR TST33+2" (726)
4065                                     ;GET ADDR OF INSTR UNDER TCE E44 IS BAD (AFIR53(1))*R/CLA
4066 :*          IF BEN15 FAILS EXECUTION WILL GO TO STATE D12.60.
4067 :*
4068 :*          IF B FORK FAILS (AFTER STATE D12.10) CAUSED BY IRCB
4069 :*          K/CLASS STUCK LOW EXECUTION WILL GO TO STATE RSD.00 CAUSING A TRAP TO 10.
4070 :*          IF IRCB B1 RAB00 IS STUCK HIGH EXECUTION WILL GO FROM
4071 :*          D12.10 TO JSR.40. THIS WILL CAUSE THE PROCESSOR TO HANG.
4072 :*          IF EITHER GRAB OBD(1) IS STUCK HIGH OR NOT GETTING THRU TO RAEL E71,
4073 :*          EXECUTION WILL GO TO STATE D12.30.
4074 :*          IF GRAB DRMX00 IS STUCK HIGH OR GRAB E50 IS BAD, EXECUTION
4075 :*          WILL GO TO D10.60 WHICH WILL HANG THE PROCESSOR.
4076 :*
4077 :*          ROM FLOW-1,175,33
4078 :*          *****
4079 004612 005200          TST34: INC     R0          ;INCREMENT TEST NUMBER
4080 004614 012702 001164  MOV     #TMP1,R2     ;PUT ADDRESS OF $TMP1 IN R2
4081 004620 012705 177400  MOV     #177400,R5   ;PUT 177400 IN R5

```

M07

```

4082 004624 010512          MOV      R5,(R2)          ;PUT -1 IN $TMP1
4083 004626 000240          SYNC34: NOP
4084 004630                IUT34:
4085 004630 005712          TST      (R2)          ;EXECUTE INSTRUCTION UNDER TEST
4086 004632                1$:
4087 004632 100416          BMI     TST35          ;:BRANCH IF INSTRUCTION SET CC'S
4088                ;FAILURE-TRY ROM FLOW 1,175,31,132
4089 004634 012737 177777 001164  MOV     #-1,$TMP1      ;PUT -1 IN $TMP1
4090 004642 012705 001164  MOV     $TMP1,R5       ;PUT ADDR OF $TMP1 IN R5
4091 004646 005215          INC     (R5)          ;INCREMENT $TMP1
4092 004650 001401          BEQ     2$            ;BRANCH IF INC WORKED
4093 004652 000000          HALT                ;EITHER D12.10 FAILED OR BEN15 FAILED
4094                ;FOR LOOPING CHANGE TO "BR TST34+2" (760)
4095 004654 022737 000000 001164 2$:  CMP     #0,$TMP1      ;DID $TMP1 GO TO ZERO?
4096 004662 001401          BEQ     3$            ;BRANCH IF YES
4097 004664 000000          HALT                ;CANNOT DIAGNOSE ERROR
4098                ;FOR LOOPING CHANGE TO "BR TST34+2" (753)
4099 004666                3$:
4100 004666 000000          HALT                ;TST.10 FAILED
4101                ;FOR LOOPING CHANGE TO "BR TST34+2" (752)

```

```

*****
*TEST 35      THREE MICROSTATES (DAC*DM1*BIT.B*DRO(0))

```

```

*
* THIS TEST IS THE SAME AS THE PREVIOUS TEST EXCEPT A BIT INSTRUCTION IS USED.
* IF FORK A FAILS RACE BIN*SMD H FAILED.
*
* IF FORK B FAILS THE INSTRUCTION DECODE ROM WORD IS BAD.
*
* IF THE RESULTANT DATA IS BAD STATE TST.10 FAILED.

```

```

* ROM FLOW-1,175,33

```

```

*****

```

```

4113                TST35: INC     R0          ;INCREMENT TEST NUMBER
4114 004670 005200          CLR     R5            ;CLEAR R5
4115 004672 005005          MOV     $TMP1,R1     ;PUT ADDR OF $TMP1 IN R1
4116 004674 012701 001164  MOV     R5,(R1)      ;CLEAR $TMP1.
4117 004700 010511          MOV     $BIT15,R5    ;PUT 100000 IN R5
4118 004702 012705 100000  SYNC35: NOP
4119 004706 000240          IUT35:
4120 004710                BIT     R5,(R1)      ;EXECUTE INSTRUCTION UNDER TEST
4121 004710 030511          BEQ     TST36        ;:BRANCH IF CC OK
4122 004712 001401          HALT                ;STATE TST.10 FAILED
4123 004714 000000                ;FOR LOOPING CHANGE TO "BR TST35+2" (766)
4124                ;*****

```

```

*TEST 36      THREE MICROSTATES (DAC*DM1*CMP.B*DRO(0))

```

```

*
* THIS TEST IS THE SAME AS THE PREVIOUS TWO TESTS
* EXCEPT A CMP INSTRUCTION IS USED.

```

```

* ROM FLOW-1,175,33

```

```

*****

```

```

4132                TST36: INC     R0          ;INCREMENT TEST NUMBER
4133 004716 005200          CLR     R5            ;CLEAR R5
4134 004720 005005          MOV     $TMP1,R1     ;PUT ADDR OF $TMP1 IN R1
4135 004722 012701 001164  MOV     R5,(R1)      ;CLEAR $TMP1
4136 004726 010511          SYNC36:
4137 004730

```

```

4138 004730 000244          CLZ          ;ENSURE Z CLEAR
4139 004732          IUT36:          ;EXECUTE INSTRUCTION UNDER TEST
4140 004732 020511          CMP      R5,(R1)      ;BRANCH IF CC OK
4141 004734 001401          BEQ      TST37       ;STATE TST.10 FAILED
4142 004736 000000          HALT           ;FOR LOOPING CHANGE TO "BR TST36+2" (770)
4143
4144 ;*****
4145 ;*TEST 37      THREE MICROSTATES (DAC*DM2*TST.B*DRO(0))
4146 ;*
4147 ;*      IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TRAP TO 10.
4148 ;*      THIS WILL ONLY HAPPEN IF RACE E45 IS BAD (AFIRO4(1)*R/CLASS).
4149 ;*
4150 ;*      BENIS & FORK B HAVE ALREADY BEEN TESTED
4151 ;*      IF THE AUTO INC FAILS IT WILL BE DUE TO A BAD
4152 ;*      FIELD IN ROM STATE D12.10.
4153 ;*
4154 ;*      ROM FLOW-2,175,33
4155 ;*
4156 004740 005200          TST37: INC      R0          ;INCREMENT TEST NUMBER
4157 004742 005001          CLR      R1          ;CLEAR R1
4158 004744 012705 001164          MOV     #STMP1,R5     ;PUT ADDR. OF STMP1 IN R5
4159 004750 010115          MOV     R1,(R5)      ;CLEAR STMP1
4160 004752 000240          SYNC37: NOP
4161 004754          IUT37:
4162 004754 005725          TST     (R5)+        ;EXECUTE INSTR. UNDER TEST
4163 004756 001006          BNE     1$          ;BRANCH IF BAD CC
4164 004760 022705 001166          CMP     #STMP1+2,R5  ;DID R5 AUTO INC?
4165 004764 001407          BEQ     TST40       ;BRANCH IF TEST OK
4166 004766 010567 174162          MOV     R5,$REGO    ;SAVE REG 0 FOR TYPEOUT
4167 004772 000000          HALT           ;NO AUTO INC STATE D12.10 BAD
4168 ;*
4169 004774 013767 177776 174174 1$: MOV     @#PSW,$ERPSW  ;FOR LOOPING CHANGE TO "BR TST37+2" (763)
4170 005002 000000          HALT           ;SAVE PSW FOR TYPEOUT
4171 ;*
4172 ;*      ;BAD CC
4173 ;*      ;FOR LOOPING CHANGE TO "BR TST37+2" (757)
4174 ;*
4175 ;*
4176 ;*
4177 ;*
4178 ;*
4179 ;*
4180 ;*
4181 ;*
4182 ;*
4183 ;*
4184 ;*
4185 ;*
4186 ;*
4187 ;*
4188 ;*
4189 ;*
4190 ;*
4191 ;*
4192 ;*
4193 ;*

```

```

4194
4195
4196
4197
4198
4199
4200
4201
4202 005004 005200
4203 005006 005001
4204 005010 012705 005032
4205 005014 010502
4206 005016 000240
4207 005020
4208 005020 000115
4209 005022 020205
4210 005024 001001
4211 005026 000000
4212
4213 005030
4214 005030 000000
4215
4216 005032 005701
4217 005034 001403
4218 005036 062706 000002
4219 005042 000000
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229 005044 005200
4230 005046 012705 005056
4231 005052 000240
4232 005054
4233 005054 000125
4234 005056 022705 005060
4235 005062 001401
4236 005064 000000
4237
4238
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248 005066 005200
4249 005070 012701 005102

```

```

* FUP.00.
* IF IRCB FJ CLASS IS STUCK HIGH EXECUTION WILL GO TO
* STATE D12.00 AND THE JMP WON'T JUMP.
*
* IF THE INSTRUCTION FAILS TO JUMP, STATE JMP.00 IS REPORTED AS BAD.
*
* ROM FLOW-1,135,35
* *****
TST40: INC R0 ;INCREMENT TEST NUMBER
CLR R1 ;ENSURE R1 CLEAR
MOV #JMPIADR,R5 ;PUT JMP ADDR. IN R5
MOV R5,R2 ;PUT JUMP ADDR IN R2
SYNC40: NOP
IUT40:
JMP (R5) ;EXECUTE INSTRUCTION UNDER TEST
CMP R2,R5 ;DID NEGATE OCCUR?
BNE ZS ;BRANCH IF YES
HALT ;STATE JMP.00 DID NOT LOAD PCB OR BENIS FAILED
;FOR LOOPING CHANGE TO "BR TST40+2" (767)
ZS:
HALT ;IRCB FJ/CLASS NOT GETTING THRU B FORK MUX
;FOR LOOPING CHANGE TO "BR TST40+2" (766)
JMPIADR: TST R1 ;IS R1 STILL ZERO?
BEQ TST41 ;BRANCH IF YES
ADD #2,SP ;JSR OCCURRED RE-ADJUST SP
HALT ;RACB IR(14:9)04 STUCK LOW
;FOR LOOPING CHANGE TO "BR TST40+2" (761)
* *****
* TEST 41 THREE MICROSTATES (JMP*DM2)
*
* THIS TEST IS THE SAME AS THE PREVIOUS TEST EXCEPT THE DM=2.
* IF FORK A FAILS RAC E45 IS BAD (AFIRO4(1)*[JMP+JSR+SWAB]).
*
* ROM FLOW-2,135,35
* *****
TST41: INC R0 ;INCREMENT TEST NUMBER
MOV #JMP2ADR,R5 ;PUT ADDRESS OF 1$ IN R5
SYNC41: NOP
IUT41:
JMP (R5)+ ;EXECUTE INSTRUCTION UNDER TEST
JMP2ADR: CMP #JMP2ADR+2,R5 ;DID R5 AUTO INC?
BEQ TST42 ;BRANCH IF YES
HALT ;NO AUTO INC
;FOR LOOPING CHANGE TO "BR TST41+2" (770)
* *****
* TEST 42 THREE MICROSTATES (JMP*DM4)
*
* IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TRAP TO 10.
* THIS WILL ONLY HAPPEN IF RACE E33(AFIRO5(1)*[JMP+JSR+SWAB]) IS BAD.
*
* ALL OTHER LOGIC HAS BEEN TESTED.
*
* ROM FLOW-4,122,35
* *****
TST42: INC R0 ;INCREMENT TEST NUMBER
MOV #NEXTT,R1 ;PUT ADDRESS OF 1$+2 IN R1

```

COB

```

4250 005074 000240 SYNC42: NOP
4251 005076 IUT42:
4252 005076 000141 JMP -(R1) ;EXECUTE INSTRUCTION UNDER TEST
4253 005100 000240 NOP
4254 005102 000240 NEXTT: NOP
4255 *****
4256 :TEST 43 THREE MICROSTATES (SOB)
4257 *
4258 * IF RACH U/CLASS IS NOT GOING HIGH EXECUTION WILL GO TO RSD.00
4259 * CAUSING A TRAP TO LOCATION 10 WITH THE ADDRESS OF 5$-2
4260 * ON THE TOP OF THE STACK.
4261 * IF EITHER RACE E10 IS BAD OR RACE BIN DOES NOT GO HIGH
4262 * EXECUTION WILL GO TO D67.01. EXECUTION WILL EVENTUALL GO
4263 * TO RSD.00 AFTER STATE D10.60 AND THE STACKED PC WILL BE AT 5$.
4264 * IF RACF EB FAILS EXECUTION WILL GO TO ASC.10 WHICH WILL
4265 * PERFORM AN ASHC*DMU OPERATION.
4266 *
4267 * IF THE SOB BRANCHES WHEN IT IS NOT SUPPOSE TO EITHER
4268 * GRAE SR EQ ONE IS STUCK HIGH OR RACK E63 IS BAD
4269 * OR STATE SOB.10 DOES NOT RESTORE THE OLD PC.
4270 *
4271 * IF THE SOB DOES NOT BRANCH WHEN IT IS SUPPOSE TO EITHER
4272 * STATE SOB.00 IS BAD OR GRAE SR EQ ONE STUCK LOW OR
4273 * RACK E63(C1) IS BAD.
4274 *
4275 * IF THE REGISTER DOES NOT DECREMENT STATE SOB.20 IS BAD.
4276 *
4277 * ROM FLOW-57,242/262.262
4278 *****
4279 005104 005200 001100 TST43: INC R0 ;INCREMENT TEST NUMBER
4280 005106 012706 007777 MOV #STACK,SP ;INITIALIZE THE SP
4281 005112 012705 MOV #7777,R5 ;SETUP R5
4282 005116 000401 BR SKIP ;SKIP NEXT INSTRUCTION
4283 005120 000410 GOOD: BR BAD+2 ;LOCATION FOR SOB TO BRANCH TO
4284 005122 005004 SKIP: CLR R4 ;SETUP R4
4285 005124 000240 SYNC43: NOP
4286 005126 IUT43:
4287 005126 077404 SOB R4,GOOD ;EXECUTE SOB WITH BRANCH
4288 005130 005705 TST R5 ;DID R5 CLEAR?
4289 005132 001001 5$: BNE 3$ ;BRANCH IF NO
4290 005134 000000 HALT ;RACF EB IS BAD(BUF AFIR11(1)*U/CLASS)
4291 ;FOR LOOPING CHANGE TO "BR TST43+2" (764)
4292 005136 3$:
4293 005136 000000 HALT ;EITHER GRAE SR EQ ONE IS STUCK LOW
4294 ;OR RACK E63(C1) IS BAD OR SOB.10 IS BAD
4295 ;FOR LOOPING CHANGE TO "BR TST43+2" (763)
4296 005140 BAD:
4297 005140 000000 HALT ;SOB FAILED TO BRANCH. SOB.00 BAD
4298 ;EITHER GRAE SR EQ ONE STUCK HIGH OR
4299 ;RACK E63(C1) BAD OR SOB.00 BAD
4300 ;FOR LOOPING CHANGE TO "BR TST43+2" (762)
4301 005142 005005 CLR R5 ;SET UP R5 FOR
4302 005144 005205 INC R5 ;NO BRANCH CONDITION
4303 005146 000240 SYNC43: NOP
4304 005150 IUT43:
4305 005150 077505 SOB R5,BAD ;EXECUTE SOB WITHOUT BRANCH
    
```

```

4306 005152 005705          TST      R5          ;DID R5 DECREMENT?
4307 005154 001401          BEQ      1$          ;BRANCH IF YES
4308 005156 000000          HALT                    ;R5 DID NOT DECREMENT. SOB.20 BAD
4309                                     ;FOR LOOPING CHANGE TO "BR BAD+2+2" (772)
4310 005160 005005          1$: CLR      R5
4311 005162 005001          CLR      R1
4312 005164 005201          2$: INC      R1
4313 005166 077502          SOB      R5,2$        ;CHECK ALL PATTERNS OF R5 FOR SOB
4314 005170 005705          TST      R5          ;DID R5 GET ALL THE WAY BACK TO 0?
4315 005172 001002          BNE      3$          ;BRANCH IF NO
4316 005174 005701          TST      R1          ;DID R1 ROLL OVER?
4317 005176 001401          BEQ      TST44        ;BRANCH IF YES
4318 005200
4319 005200 000000          3$: HALT              ;THE SIGNAL "SR EQ ONE" FAILED
4320                                     ;FOR LOOPING CHANGE TO "BR 1$" (767)
4321 .SBTTL

```

```

*****
*TEST 44          FOUR MICROSTATES (DAC*DM12*P/CLASS*DRO(0)
*
* IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TRAP TO 10.
* THIS WILL ONLY HAPPEN IF RACJ AFIR 14(1) DOES NOT
* GET THRU RAC E42.
*
* THE FOLLOWING COULD BE FORK B FAILURES:
* IF IRCB 80 RAB00 IS STUCK HIGH OR NOT GETTING THRU
* TO RAC. RADR00 EXECUTION WILL GO TO EXC.90 WHICH WOULD
* PUT THE RESULT INTO A REGISTER RATHER THAN MEMORY.
* IF IRCB E38(6) IS STUCK HIGH EXECUTION WILL GO TO
* RSD.00 CAUSING A TRAP TO 10.
* IF THE BIC DOES NOT HAPPEN THEN EXC.00 IS BAD.

```

```

ROM FLOW-2,175,31,132
*****
4339 005202 005200          TST44: INC      RC          ;INCREMENT TEST NUMBER
4340 005204 012701 001164   MOV      #STMP1,R1      ;PUT ADDRESS OF STMP1 IN R1
4341 005210 005005          CLR      R5            ;PUT A 1
4342 005212 005205          INC      R5            ;IN R5
4343 005214 010511          MOV      R5,(R1)       ;PUT A 1 IN STMP1
4344 005216 000240          SYNC44: NOP
4345 005220          TUT44:
4346 005220 040521          BIC      R5,(R1)+      ;EXECUTE INSTR. UNDER TEST
4347 005222 022701 001156   CMP      #STMP1+2,R1   ;DID R1 AUTO INC?
4348 005226 001404          BEQ      1$            ;BRANCH IF YES
4349 005230 005701          TST      R1            ;DID INSTR GO THRU EXC.90?
4350 005232 001001          BNE      2$            ;BRANCH IF NO
4351 005234 000000          HALT                    ;EITHER IRCB 80 RAB00 IS STUCK HIGH
4352                                     ;OR IT IS NOT GETTING THRU TO RACL RADR50
4353                                     ;FOR LOOPING CHANGE TO "BR TST44+2" (763)
4354 005236          2$:
4355 005236 000000          HALT                    ;INSTRUCTION FAILED. CAN'T DETERMINE CAUSE
4356                                     ;FOR LOOPING CHANGE TO "BR TST44+2" (762)
4357 005240 012701 001164   1$: MOV      #STMP1,R1   ;PUT ADDR OF STMP1 IN R1
4358 005244 005711          TST      (R1)          ;DID BIC WORK?
4359 005246 001401          BEQ      3$            ;BRANCH IF YES
4360 005250 000000          HALT                    ;STATE EXC.00 FAILED
4361                                     ;FOR LOOPING CHANGE TO "BR TST44+2" (755)

```

E08

```

4362 005252 010511          35:  MOV    R5,(R1)      ;PUT 1 IN $TMP2 & CLEAR Z
4363 005254 040521          BIC    R5,(R1)+    ;EXECUTE INSTRUCTION UNDER TEST
4364 005256 001401          BEQ    TST45       ;CC'S OK
4365 005260 000000          HALT              ;STATE EXC.00 BAD
4366                                     ;FOR LOOPING CHANGE TO "BR TST44+2" (751)
4367 :*****
4368 :TEST 45      FOUR MICROSTATES (DAC*DM12*TST.B*DRO(1))
4369 :
4370 :      AFTER STATE D12.10 IF EITHER GRAB OBD(1) DOES NOT GO HIGH
4371 :      OR DOES NOT GET THRU RACL E71 EXECUTION WILL GO
4372 :      TO TST.10. IF EITHER GRAB OBD(0) IS STUCK HIGH OR NOT GETTING
4373 :      THRU RACK E41, EXECUTION WILL GO TO D10.60. THIS WILL CAUSE
4374 :      THE PROCESSOR TO HANG UP IN THE PAUSE STATE AT MICRO
4375 :      ADDRESS 177. IF THE TEST FAILS THEN STATE D12.30 FAILED.
4376 :
4377 :      ROM FLOW-1,175,137,33
4378 :*****
4379 005262 005200          TST45: INC    R0          ;INCREMENT TEST NUMBER
4380 005264 012705 001164  MOV    #TMP1,R5     ;PUT ADDRESS 00 $TMP1 IN R5
4381 005270 012701 100000  MOV    #BIT15,R1    ;PUT NEGATIVE UPPER BYTE IN R1
4382 005274 010115          MOV    R1,(R5)     ;MOVE R1 TO $TMP1
4383 005276 005205          INC    R5          ;PUT ODD ADDR IN R5
4384 005300 000240          SYNC45: NOP
4385 005302          IUT45:
4386 005302 105715          TSTB   (R5)        ;EXECUTE INSTR UNDER TEST
4387 005304 100401          BMI   TST46       ;TEST OK, GO TO NEXT TEST
4388 005306 000000          HALT              ;EITHER STATE D12.30 FAILED OR
4389                                     ;BEN05*FEN2 FAILED(SEE ABOVE)
4390                                     ;FOR LOOPING CHANGE TO "BR TST45+2" (766)
4391 :*****
4392 :TEST 46      FOUR MICROSTATES (DAC*DM4*TST.B*DRO(0))
4393 :
4394 :      IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TRAP TO 10.
4395 :      THIS WILL ONLY OCCUR IF RACE E33(AFIROS(1)*R/CLASS) IS BAD.
4396 :
4397 :      AFTER D10.30 IF IRCB FJ/CLASS DOES NOT GET TO RACL E71
4398 :      OR IF RACL E71 IS BAD EXECUTION WILL GO TO SVC.50.
4399 :
4400 :      IF THE INSTRUCTION DOESN'T WORK THEN D10.60 IS BAD.
4401 :
4402 :      ROM FLOW-4,122,177,33
4403 :*****
4404 005310 005200          TST46: INC    R0          ;INCREMENT TEST NUMBER
4405 005312 012701 100000  MOV    #BIT15,R1    ;SET SIGN BIT
4406 005316 012705 001164  MOV    #TMP1,R5     ;PUT ADDR OF $TMP1 IN R5
4407 005322 010125          MOV    R1,(R5)+    ;SET SIGN BIT IN $TMP1
4408 005324 005015          CLR   (R5)        ;ENSURE $TEMP2 CLEAR
4409 005326 000240          SYNC46: NOP
4410          IUT46:
4411 005330 005745          TST   -(R5)       ;EXECUTE INSTRUCTION UNDER TEST
4412 005332 100407          BMI   TST47       ;TEST OK, GO TO NEXT TEST
4413 005334 022706 001160  CMP    #TMP1-4,SP   ;DID EXECUTION GO TO SVC.50?
4414 005340 001003          BNE   IS          ;BRANCH IF NO
4415 005342 012706 001100  MOV    #STACK,SP   ;RESTORE THE SP
4416 005346 000000          HALT              ;BEN15*FEN2 FAILED(SEE ABOVE)
4417                                     ;FOR LOOPING CHANGE TO "BR TST46+2" (761)
  
```


4418 005350
4419 005350 000000
4420
4421
4422
4423
4424
4425
4426
4427
4428
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4440
4441
4442
4443
4444
4445
4446
4447 005352 005200
4448 005354 005306
4449 005356 005306
4450 005360 012705 000340
4451 005364 010516
4452 005366 005306
4453 005370 005306
4454 005372 012705 005432
4455 005376 010516
4456 005400 012705 001164
4457 005404 005015
4458 005406 012701 100000
4459 005412 000240
4460 005414
4461 005414 010165 000002
4462 005420 012706 001100
4463 005424 005715
4464 005426 100002
4465 005430 000000
4466
4467 005432
4468 005432 000000
4469
4470
4471
4472
4473

15: HALT ; STATE D10.60 FAILED
; FOR LOOPING CHANGE TO "BR TST46+2" (760)

*THE LOGICAL SEQUENCE HERE WOULD BE TO TEST THE BIT & CMP INSTRUCTIONS
*BUT NO ADDITIONAL LOGIC IS TESTED BY THEM.

*TEST 47 FOUR MICROSTATES (DAC*DM6*0/CLASS)

* FORK A SHOULD NOT FAIL ON THIS TEST.
* BEND1 SHOULD NOT FAIL.
* BEN15*FEN2 SHOULD NOT FAIL.
* IF D67.00 FAILS TO INCREMENT THE PC AN RTI INSTRUCTION
* WILL BE EXECUTED.
* IF D67.00 FAILS TO CLOCK THE BR THE INSTRUCTION
* WILL BE ADDED AS THE INDEX WORD.
* IF D67.10 FAILS TO ADD THE INDEX NUMBER THE SOURCE
* WILL BE PUT IN THE WRONG LOCATION.

* ROM FLOW-6,251,122,157

TST47: INC R0 ; INCREMENT TEST NUMBER
DEC SP ; THIS
DEC SP ; GROUP
MOV #340,R5 ; OF INSTRUCTIONS
MOV R5,(SP) ; SETS UP THE
DEC SP ; STACK
DEC SP ; TO HANDLE
MOV #BAD1,R5 ; AN ERONEOUS
MOV R5,(SP) ; RTI INSTRUCTION
MOV #STMP1,R5 ; PUT ADDR OF STMP1 IN R5
CLR (R5) ; CLEAR STMP1
MOV #BIT15,R1 ; SET SIGN BIT IN R1

SYNCH7: NOP
IUT47: MOV R1,2(R5) ; EXECUTE INSTRUCTION UNDER TEST
MOV #STACK,SP ; RESTORE THE SP
TST (R5) ; DID INDEX WORD GET ADDED?
BPL TST50 ; BRANCH IF YES
HALT ; D67.10 FAILED TO ADD INDEX
; FOR LOOPING CHANGE TO "BR TST47+2" (751)

BAD1: HALT ; D67.00 FAILED TO INC PC
; FOR LOOPING CHANGE TO "BR TST47+2" (750)

*TEST 50 FOUR MICROSTATES (BIN*SM12*DM0*-DF7*SRO(1))

* IF FORK A FAILS EXECUTION WILL GO TO STATE D12.00.

G08

4474
4475
4476
4477
4478
4479
4480
4481
4482
4483
4484
4485
4486
4487 005434 005200
4488 005436 012705 001164
4489 005442 012701 100000
4490 005446 010115
4491 005450 005205
4492 005452 012701 000200
4493 005456 000240
4494 005460
4495 005460 131501
4496 005462 100405
4497 005464 010215
4498 005466 131501
4499 005470 100401
4500 005472 000000
4501
4502
4503
4504 005474
4505 005474 000000
4506
4507
4508
4509
4510
4511
4512
4513
4514
4515
4516
4517
4518
4519
4520
4521
4522
4523
4524
4525
4526
4527
4528
4529 005476 005200

THIS WILL HAPPEN IF RACE BF1=7 DOES NOT GO HIGH.
STATE D12.00 WOULD BITB R5 & THE CONTENTS OF LOCATION 200 WHICH IS 137.
THE FOLLOWING COULD BE BEN14*FORK C FAILURES:
IF IRCC CO RAB00 DOES NOT GO HIGH EXECUTION WILL GO TO 000.90 WHICH WILL TEST THE LOW BYTE INSTEAD OF THE HIGH BYTE.
IF STATE 000.80 FAILS TO SWAP THE BYTES IT WILL LOOK LIKE A FORK C FAILURE.
ROM FLOW-21,27,204,205

TST50: INC R0 ;INCREMENT TEST NUMBER
MOV #STMP1,R5 ;PUT ADDRESS OF STMP1 IN R5
MOV #BIT15,R1 ;PUT NEG HIGH & POS LOW BYTE IN R1
MOV R1,(R5) ;PUT IN STMP1
INC R5 ;PUT ADDR OF STMP1 H BYTE IN R5
MOV #200,R1 ;PUT POS HIGH & NEG LOW BYTE IN R1
SYNCS0: NOP
IUT50: BITB (R5),R1 ;EXECUTE INSTRUCTION UNDER TEST
BMI TST51 ;TEST OK, GO TO NEXT TEST
MOV R2,(R5) ;PUT 200 IN STMP1
BITB (R5),R1 ;EXECUTE FAILED INSTRUCTION
BMI IS ;BRANCH IF FORK C FAILED
HALT ;RACE BF1=7 NOT GOING HIGH
;EITHER RACJ AFIR14(1) DOES NOT
;GET TO RACE E40 OR E40 BAD
;FOR LOOPING CHANGE TO "BR TST50+2" (751)
IS: HALT ;EITHER IRCC CO RAB00 DOES NOT GO HIGH
;OR STATE 000.80 FAILED TO SWAP THE BYTES
;FOR LOOPING CHANGE TO "BR TST50+2" (760)

*TEST 51 FOUR MICROSTATES (BIN*SM12*DMO*DF7*SR0(0))
*
* IF FORK A FAILS EXECUTION WILL EITHER GO TO STATE D12.00 OR EXC.80.
* STATE D12.00 WOULD ADD R5 TO THE CONTENTS OF THE PC.
* STATE EXC.80 WOULD NOT CHANGE THE PC.
*
* IF FORK C FAILS EXECUTION WILL EITHER GO TO JSR.10 OR ASC.80.
* JSR.10 WILL CAUSE R5 TO BE STACKED, THE PC TO BE PUT
* IN R5, AND THE PC REPLACED BY R5 WHICH WILL CAUSE AND RTI SINCE
* THE CONTENTS OF THE LOCATION POINTED TO BY R5 IS 000002.
* ASC.80 WILL CAUSE THE ADD TO LOOK LIKE IT FAILED.
*
* IF STATE 007.10 FAILS TO LOAD THE SHFTR THE PC WILL
* DOUBLE AND THE PROGRAM WILL BLOW UP.
*
* IF THE SHFTR FAILS TO BE PUT IN THE SR THE PC
* WILL NOT CHANGE.
*
* ROM FLOW-21,27,203,30

TST51: INC R0 ;INCREMENT TEST NUMBER

```

4530 005500 012706 001100      MOV      #STACK,SP      ;INITIALIZE SP
4531 005504 012701 000340      MOV      #340,R1        ;PUT PRIORITY LEVEL 7 IN R5
4532 005510 010146              MOV      R1,-(SP)        ;PUT ON STACK
4533 005512 012701 005610      MOV      #SRTI,R1       ;PUT RETURN ADDR IN R1
4534 005516 010146              MOV      R1,-(SP)        ;PUT ON STACK FOR FORK C FAILURE
4535 005520 005306              DEC      SP              ;ADJUST THE
4536 005522 005306              DEC      SP              ;SP
4537 005524 005005              CLR      R5              ;PUT ADDRESS 0 IN R5
4538 005526 012701 000002      MOV      #2,R1          ;PUT OFFSET IN R1
4539 005532 010115              MOV      R1,(R5)        ;PUT OFFSET IN LOCATION 0
4540 005534 000240              SYNC51: NOP
4541 005536              IUT51:
4542 005536 061507              ADD      (R5),PC        ;EXECUTE INSTRUCTION UNDER TEST
4543 005540 000401              BR       6$             ;EITHER FORK A OR FORK C OR D07.10 FAILED
4544 005542 000423              BR       TST52          ;:TEST OK GO TO NEXT TEST
4545 005544 012705 000004      6$:      MOV      #4,R5   ;SET BIT 2 IN R5
4546 005550 061507              ADD      (R5),PC        ;EXECUTE FAILED INSTR.
4547 005552 000261      3$:      SEC              ;WILL CHANGE TO SEC!SEZ IF THE INSTR GOES
4548              ;THRU D12.00. STATE EXC.8 OR ASC.8
4549              ;WOULD NOT SET Z WHILE A FAILURE OF
4550              ;STATE D07.10 WILL CAUSE THE ERROR
4551              ;1$-2 TO REPORT.
4552 005554 000401              BR       1$             ;SKIP NEXT INSTRUCTION
4553 005556 000000              HALT                    ;STATE D07.10 DID NOT LOAD SR
4554              ;FOR LOOPING CHANGE TO "BR TST51+2" (750)
4555 005560 001004      1$:      BNE      2$             ;BRANCH IF Z DID NOT SET
4556 005562 012767 000261 177762      MOV      #261,3$        ;RESTORE ORIGINAL VALUE OF LOCATION 3$
4557 005570 000000              HALT                    ;RACE BF1=7 DID NOT GO HIGH
4558              ;FOR LOOPING CHANGE TO "BR TST51+2" (743)
4559 005572 012705 100000      2$:      MOV      #BIT15,R5   ;SET SIGN BIT IN R5
4560 005576 000250              CLN                      ;ENSURE N CLEAR
4561 005600 061507              ADD      (R5),PC        ;EXECUTE FAILED INSTR
4562 005602 100401              BMI      4$             ;BRANCH IF ADD OCCURED IN STATE EXC.80
4563 005604 000000              HALT                    ;EITHER IRCC CD RAB02 IS BEING HELD LOW
4564              ;OR IT IS NOT GETTING THRU
4565              ;TO RACL RAD02
4566              ;FOR LOOPING CHANGE TO "BR TST51+2" (735)
4567 005606      4$:      HALT                    ;RACE BIN IS NOT GOING LOW
4568 005606 000000              HALT                    ;FOR LOOPING CHANGE TO "BR TST51+2" (734)
4569
4570 005610      SRTI:
4571 005610 000000              HALT                    ;EITHER IRCC CD RAB01 IS STUCK HIGH
4572              ;OR IRC E40 IS BAD OR IRC E40(14)
4573              ;IS STUCK HIGH
4574              ;OR CD RAB01 IS NOT GETTING THRU TO RACL RAD01
4575              ;FOR LOOPING CHANGE TO "BR TST51+2" (733)
4576
4577      ;*****
4578      ;*TEST 52      FOUR MICROSTATES (BIN*SM12*DMO*DF7*SRO(1))
4579      ;*
4580      ;*      IF FORK A FAILS EXECUTION WILL GO TO D12.00.
4581      ;*
4582      ;*      FORK C SHOULD NOT FAIL SINCE THE LOGIC HAS ALREADY BEEN TESTED.
4583      ;*
4584      ;*      IF STATE D07.00 FAILS TO SWAP THE BYTES THE CC'S WILL
4585      ;*      BE BAD.
4586      ;*      IF D07.00 FAILS TO LOAD THE SHIFTER, THE THIRD CMPB WILL FAIL.

```

```

4586 ;* IF D07.00 FAILS TO LOAD THE SR THE SECOND CMPB WILL FAIL.
4587 ;*
4588 ;* ROM FLOW-21,27,202,30
4589 ;* *****
4590 005612 005200 TST52: INC R0 ; INCREMENT TEST NUMBER
4591 005614 005005 CLR R5 ; PUT ADDRESS 0 IN R5
4592 005616 012701 005634 MOV #POINT,R1 ; PUT ADDR OF POINT IN R1
4593 005622 000301 SWAB R1 ; EXCHANGE BYTES
4594 005624 010115 MOV R1,(R5) ; PUT DATA IN LOCATION 0
4595 005626 005205 INC R5 ; CHANGE R5 TO HIGH BYTE ADDRESS
4596 005630 000240 SYNC52: NOP
4597 005632 IUT52:
4598 005632 121507 POINT: CMPB (R5),PC ; EXECUTE INSTRUCTION UNDER TEST
4599 005634 001406 BEQ 4$ ; BRANCH IF OK
4600 005636 016705 000002 MOV 2$,R5 ; PUT CONTENTS OF 2$ IN R5
4601 005642 121507 CMPB (R5),PC ; EXECUTE FAILED INSTRUCTION
4602 005644 001401 2$: BEQ 3$ ; BRANCH IF FORK A FAILED
4603 005646 000000 HALT ; STATE D07.00 FAILED TO SWAB OR LOAD SR
4604 ; FOR LOOPING CHANGE TO "BR TST52+2" (762)
4605 005650 3$:
4606 005650 000000 HALT ; RACE BF1=0 NOT GOING HIGH
4607 ; FOR LOOPING CHANGE TO "BR TST52+2" (761)
4608 005652 121507 4$: CMPB (R5),PC ; DID SHFTR GET LOADED
4609 005654 001001 BNE TST53 ; BRANCH IF YES
4610 005656 000000 HALT ; D07.00 DID NOT LOAD SHFTR
4611 ; FOR LOOPING CHANGE TO "BR TST52+2" (756)
4612
4613
4614
4615
4616
4617
4618
4619
4620
4621
4622
4623
4624
4625
4626 005660 005200 TST53: INC R0 ; INCREMENT TEST NUMBER
4627 005662 012705 001164 MOV #STMP1,R5 ; PUT ADDRESS OF STMP1 IN R5
4628 005666 012701 100000 MOV #BIT15,R1 ; SET SIGN BIT IN R1
4629 005672 010125 MOV R1,(R5)+ ; SET SIGN BIT STMP1 & STEP R5 TO STMP2
4630 005674 005015 CLR (R5) ; CLEAR STMP2
4631 005676 012701 000002 MOV #2,R1 ; PUT ADDRESS OF LOC 2 IN R1
4632 005702 005011 CLR (R1) ; CLEAR LOCATION ZERO
4633 005704 000240 SYNC53: NOP
4634 005706 IUT53:
4635 005706 054501 BIS -(R5),R1 ; EXECUTE INSTRUCTION UNDER TEST
4636 005710 100411 BMI TST54 ; TEST OK, GO TO NEXT TEST
4637 005712 020537 000002 CMP R5,#2 ; DID FORK A FAIL?
4638 005716 001001 BNE 1$ ; BRANCH IF NO
4639 005720 000000 HALT ; EITHER RACE BF1=7 OR SMO DID NOT GO HIGH
4640 ; FOR LOOPING CHANGE TO "BR TST53+2" (760)
4641 005722 020527 001166 1$: CMP R5,#STMP2 ; DID R5 FAIL TO DECREMENT?

```

```

*****
TEST 53 FOUR MICROSTATES (BIN*SM4*DMO*-DF7*SRO(0))
*****

```

```

IF FORK A FAILS EXECUTION WILL
GO TO D45.00 WHICH WILL EXECUTE A SMO*DM4 INSTRUCTION EXCEPT
THE DESTINATION REGISTER WILL NOT DECREMENT.

FORK C WILL NOT FAIL SINCE IT HAS ALREADY BEEN TESTED.

IF THE SRC FAILS TO AUTO DECREMENT STATE S45.00 IS BAD.

```

```

*****
ROM FLOW-24,23,27,205
*****

```

```

TST53: INC R0 ; INCREMENT TEST NUMBER
MOV #STMP1,R5 ; PUT ADDRESS OF STMP1 IN R5
MOV #BIT15,R1 ; SET SIGN BIT IN R1
MOV R1,(R5)+ ; SET SIGN BIT STMP1 & STEP R5 TO STMP2
CLR (R5) ; CLEAR STMP2
MOV #2,R1 ; PUT ADDRESS OF LOC 2 IN R1
CLR (R1) ; CLEAR LOCATION ZERO
SYNC53: NOP
IUT53:
BIS -(R5),R1 ; EXECUTE INSTRUCTION UNDER TEST
BMI TST54 ; TEST OK, GO TO NEXT TEST
CMP R5,#2 ; DID FORK A FAIL?
BNE 1$ ; BRANCH IF NO
HALT ; EITHER RACE BF1=7 OR SMO DID NOT GO HIGH
; FOR LOOPING CHANGE TO "BR TST53+2" (760)
1$: CMP R5,#STMP2 ; DID R5 FAIL TO DECREMENT?

```

J08

PDP 11/70 CPU DIAGNOSTIC PART 1 MACY11 27(732) C3-NOV-76 11:18 PAGE 101
DEKBAB.P11 T53 FOUR MICROSTATES (BIN*SM4*DM0*-DF7*SRO(0))

```
4642 005726 001001          BNE      2$          ;BRANCH IF NO
4643 005730 000000          HALT           ;STATE S45.00 DID NOT DECREMENT R5
4644                                     ;FOR LOOPING CHANGE TO "BR TST53+2" (754)
4645 005732          2$:          HALT           ;INSTRUCTION FAILED
4646 005732 000000          HALT           ;FOR LOOPING CHANGE TO "BR TST53+2" (753)
4647                                     ;*****
4648                                     ;*TEST 54      FOUR MICROSTATES (RTS)
4649                                     ;*
4650                                     ;*      IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TRAP TO 10.
4651                                     ;*      THIS WILL ONLY HAPPEN IF RACF E3 DOES NOT GO HIGH.
4652                                     ;*
4653                                     ;*      IF THE PC OR R5 FAILS THE TEST WILL HALT.
4654                                     ;*
4655                                     ;*      ROM FLOW-40,223,224,342
4656                                     ;*      *****
4657                                     ;*
4658 005734 005200          TST54: INC      R0          ;INCREMENT TEST NUMBER
4659 005736 012706 001100          MOV      #STACK,SP      ;INITIALIZE THE STACK
4660 005742 012705 100000          MOV      #BIT15,R5      ;SET SIGN BIT IN R5
4661 005746 010546          MOV      R5, -(SP)      ;PUT R5 ON THE STACK
4662 005750 012705 005760          MOV      #1$,R5        ;PUT ADDRESS TO RETURN TO IN R5
4663 005754 000205          RTS      R5           ;EXECUTE INSTRUCTION UNDER TEST
4664 005756 000000          HALT           ;STATE RTS.00 FAILED TO PUT R5 IN THE PC
4665                                     ;*      ;FOR LOOPING CHANGE TO "BR TST54+2" (767)
4666 005760 005705          1$:  TST      R5           ;DID R5 GET THE TOP OF THE STACK?
4667 005762 100401          BMI      TST55        ;BRANCH IF YES
4668 005764 000000          HALT           ;THE RTS FAILED TO PUT THE STACK IN R5
4669                                     ;*      ;FOR LOOPING CHANGE TO "BR TST54+2" (764)
4670                                     ;*      *****
4671                                     ;*TEST 55      FOUR MICROSTATES (JMP*DM6)
4672                                     ;*
4673                                     ;*      IF FORK A FAILS EXECUTION WILL GO TO STATE D12.01.
4674                                     ;*      THIS WOULD CAUSE A JMP*DM1 TO EXECUTE.
4675                                     ;*
4676                                     ;*      NEITHER BEN01 NOR BEN15*FEN2 SHOULD FAIL SINCE THEY HAVE ALREADY BEEN
4677                                     ;*      TESTED.
4678                                     ;*
4679                                     ;*      ROM FLOW-6,251,122,35
4680                                     ;*      *****
4681 005766 005200          TST55: INC      R0          ;INCREMENT TEST NUMBER
4682 005770 012705 006002          MOV      #JMPT0,R5     ;PUT ADDRESS-2 TO JUMP TO IN R5
4683 005774 000240          SYNC55: NOP
4684 005776          IUT55:          JMP      2(R5)        ;EXECUTE INSTRUCTION UNDER TEST
4685 005776 000165 000002          JMPT0:  HALT           ;JMP*DM6 DID NOT JUMP OR THE OFFSET
4686 006002          ;DID NOT GET ADDED OR RACE E33 FAILED
4687 006002 000000          ;AND A JMP*DM1 WAS EXECUTED
4688                                     ;*      ;FOR LOOPING CHANGE TO "BR TST55+2" (772)
4689                                     ;*      .SBTTL
4690                                     ;*      *****
4691                                     ;*TEST 56      FIVE MICROSTATES (DAC*DM12*P/CLASS*DRO(1))
4692                                     ;*
4693                                     ;*      FORK A SHOULDN'T FAIL.
4694                                     ;*
4695                                     ;*      BEN15 SHOULDN'T FAIL SINCE THIS LOGIC HAS BEEN TESTED.
4696                                     ;*
4697                                     ;*
```

K08

4698
4699
4700
4701
4702
4703
4704
4705
4706 006004 005200
4707 006006 012705 001164
4708 006012 012701 040000
4709 006016 010115
4710 006020 005205
4711 006022 000240
4712 006024
4713 006024 106115
4714 006026 100401
4715 006030 000000
4716
4717
4718
4719
4720
4721
4722
4723
4724
4725
4726
4727
4728
4729 006032 005200
4730 006034 012705 001164
4731 006040 012701 001166
4732 006044 010115
4733 006046 005003
4734 006050 010311
4735 006052 012703 100000
4736 006056 000240
4737 006060
4738 006060 010335
4739 006062 100401
4740 006064 000000
4741
4742 006066 005711
4743 006070 100401
4744 006072 000000
4745
4746 006074 020501
4747 006076 001401
4748 006100 000000
4749
4750
4751
4752
4753

```
;* NEITHER SHOULD BENOS*FEN2.  
;* IF FORK B FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TRAP TO 10.  
;* THIS FAILURE WOULD BE CAUSED BY A BAD FIELD (PART PCLASS)  
;* IN THE IR DECODE ROM.  
;* ROM FLOW-1,175,137,31,132  
;*****  
TST56: INC R0 ;INCREMENT TEST NUMBER  
MOV #STMP1,R5 ;PUT ADDRESS OF STMP1 IN R5  
MOV #BIT14,R1 ;SET BIT 14 IN R1  
MOV R1,(R5) ;SET BIT 14 IN STMP1  
INC R5 ;SET R5 TO HIGH BYTE OF STMP1  
SYNC56: NOP  
IUT56: ROLB (R5) ;EXECUTE INSTRUCTION UNDER TEST  
BMI TST57 ;TEST OK, GO TO NEXT TEST  
HALT ;ROLB*DM1*DR5(1)) FAILED (BAD CC)  
;FOR LOOPING CHANGE TO "BR TST56+2" (766)  
;*****  
*TEST 57 FIVE MICROSTATES (DAC*DM3*0/CLASS)  
;* FORK A SHOULD NOT FAIL SINCE THE LOGIC HAS ALREADY BEEN TESTED.  
;* IF THE DR DOES NOT AUTO INC THEN STATE D30.10 IS BAD.  
;* IF THE CONDITION CODES ARE BAD THEN EITHER STATE D10.50  
;* DID NOT LOAD THE BR OR THE DOUBLE DEFERED DIDN'T WORK.  
;* ROM FLOW-3,221,233,311,157  
;*****  
TST57: INC R0 ;INCREMENT TEST NUMBER  
MOV #STMP1,R5 ;PUT ADDRESS OF STMP1 IN R5  
MOV #STMP2,R1 ;PUT ADDRESS OF STMP2 IN R1  
MOV R1,(R5) ;PUT ADDRESS OF STMP2 IN STMP1  
CLR R3  
MOV R3,(R1) ;CLEAR STMP2  
MOV #BIT15,R3 ;SET SIGN BIT IN R3  
SYNC57: NOP  
IUT57: MOV R3,@(R5)+ ;EXECUTE INSTRUCTION UNDER TEST  
BMI 1$ ;BRANCH IF N BIT SET  
HALT ;BAD CONDITION CODES  
;FOR LOOPING CHANGE TO "BR TST57+2" (763)  
1$: TST (R1) ;DID MOVE ACTUALLY TAKE PLACE?  
BMI 2$ ;BRANCH IF YES  
HALT ;SOURCE DID NOT GET MOVED TO DESTINATION  
;FOR LOOPING CHANGE TO "BR TST57+2" (760)  
2$: CMP R5,R1 ;DID R5 AUTO INC?  
BEQ TST60 ;BRANCH IF YES  
HALT ;STATE D30.10 DID NOT INC R5  
;FOR LOOPING CHANGE TO "BR TST57+2" (755)  
;*****  
*TEST 60 FIVE MICROSTATES (DAC*DM4*P/CLASS*DRO(0))  
;* FORK A SHOULD NOT FAIL.
```

4754
4755
4756
4757
4758
4759
4760
4761 006102 005200
4762 006104 012705 001164
4763 006110 005025
4764 006112 000270
4765 006114 000240
4766 006116
4767 006116 006745
4768 006120 005215
4769 006122 001401
4770 006124 000000

```
;*
;* IF FORK B FAILS, AFTER D10.60, EXECUTION WILL GO TO
;* RSD.00 CAUSING A TRAP TO 10. THIS WILL ONLY HAPPEN IF
;* THE IR DECODE ROM HAS A BAD FIELD (PART PCLASS).
;*
;* ROM FLOW-4,122,177,31,132
;*****
TST60: INC RO ;INCREMENT TEST NUMBER
      MOV #STMP1,R5 ;PUT ADDRESS OF STMP1 IN R5
      CLR (R5)+ ;CLEAR STMP1 & STEP R5 TO STMP2
      SEN ;SET N
SYNC60: NOP
IUT60: SXT -(R5) ;EXECUTE INSTRUCTION UNDER TEST.
      INC (R5) ;SHOULD MAKE STMP1=0
      BEQ TST61 ;BRANCH IF TEST OK
      HALT ;SXT DID NOT WORK
      ;FOR LOOPING CHANGE TO "BR TST60+2" (767)
```

4771
4772
4773
4774
4775
4776
4777
4778
4779

```
;*****
;THE LOGICAL SEQUENCE AT THIS POINT WOULD BE TO EXECUTE A DAC*DM4*[TST.B+
;BIT.B+CMPI.B]*DR0(1) INSTRUCTION FOLLOWED BY A DAC*DM6*[TST.B+BIT.B+CMPI.B]*
;DR0(0) INSTRUCTION TEST BUT NO ADDITIONAL LOGIC IS TESTED.
;*****
```

4780
4781
4782
4783
4784
4785
4786
4787
4788
4789

```
;*****
;THE LOGICAL SEQUENCE AT THIS POINT WOULD BE TO EXECUTE A BIN*SM4*DM0*-DF7*SRO(1)
;FOLLOWED BY A BIN*SM4*DM0*DF7*SRO(0)
;FOLLOWED BY A BIN*SM4*DM0*DF7*SRO(1) INSTRUCTION BUT NO ADDITIONAL LOGIC IS TESTED.
;*****
```

4790
4791
4792
4793
4794
4795
4796
4797
4798
4799

```
;*****
;TEST 61 FIVE MICROSTATES (BIN*SM6*DM0*-DF7*SRO(0))
;*
;* IF FORK A FAILS EXECUTION WILL GO TO STATE D67.00
;* WHICH WOULD EXECUTE A SMO*DM6 INSTRUCTION.
;*
;* BEN14*FEN4 SHOULD NOT FAIL SINCE IT HAS ALREADY BEEN TESTED
;*
;* ROM FLOW-26,54,141,142,205
```

4800 006126 005200
4801 006130 012705 001164
4802 006134 010565 000002
4803 006140 010501
4804 006142 000240
4805 006144
4806 006144 046501 000002
4807 006150 005701
4808 006152 001405
4809 006154 005767 173006

```
;*****
TST61: INC RO ;INCREMENT TEST NUMBER
      MOV #STMP1,R5 ;PUT ADDRESS OF STMP1 IN R5
      MOV R5,2(R5) ;PUT ADDRESS OF STMP1 IN STMP2
      MOV R5,R1 ;PUT ADDRESS OF STMP1 IN R1
SYNC61: NOP
IUT61: BIC 2(R5),R1 ;EXECUTE INSTRUCTION UNDER TEST
      TST R1 ;DID R1 GO TO ZERO?
      BEQ TST62 ;BRANCH IF YES
      TST STMP2 ;DID STMP2 GO TO ZERO?
```

MO8

```

4810 006160 001001          BNE    1$          ;BRANCH IF NO
4811 006162 000000          HALT          ;RACE BF1=0 DID NOT GO HIGH ON BIC
4812                                     ;FOR LOOPING CHANGE TO "BR TST61+2" (762)
4813 006164          1$:          ;
4814 006164 000000          HALT          ;INSTRUCTION FAILED
4815                                     ;FOR LOOPING CHANGE TO "BR TST61+2" (761)
4816                                     ;*****
4817                                     ;TEST 62      FIVE MICROSTATES (BIN*SM12*DM12*0/CLASS)
4818                                     ;*
4819                                     ;*
4820                                     ;*      IF FORK A FAILS EXECUTION WILL GO TO D12.01.THIS WOULD CAUSE R5
4821                                     ;*      TO BE WRITTEN INTO $TMP2.
4822                                     ;*
4823                                     ;*      IF FORK C FAILS EXECUTION WOULD GO TO ONE OF THE
4824                                     ;*      FOLLOWING STATES: D45.80,D30.80,FOP.00,WAT.00, D00.90, AND ASH.20.
4825                                     ;*      STATE D45.80 WOULD EXECUTE A SM2*DM4 INSTEAD OF SM2*DM1.
4826                                     ;*      STATE D30.80 WOULD EXECUTE A SM1*DM3.
4827                                     ;*      STATE FOP.00 WOULD CAUSE A TRAP TO LOCATION 10.
4828                                     ;*      THIS WILL ONLY HAPPEN IF EITHER IRCC CO RAB03 IS STUCK
4829                                     ;*      OR IT IS NOT GETTING THRU RAEL RAD03 OR
4830                                     ;*      IRCC FORK C MUX INPUT B0 IS HIGH.
4831                                     ;*      THE LA30 PRINTER BUFFER WILL BE SET UP TO GENERATE AN INTERRUPT IN
4832                                     ;*      CASE THE TEST FAILS TO THE WAT.00 STATE.
4833                                     ;*      STATE D00.90 WILL EXECUTE A DMO INSTEAD OF A DM1.
4834                                     ;*      STATE ASH.20 WILL CLEAR THE C BIT.
4835                                     ;*
4836                                     ;*      ROM FLOW-22,27,111,155,312
4837                                     ;*      *****
4838 006166 005200          TST62: INC    R0          ;INCREMENT TEST NUMBER
4839 006170 016767 172746 000130  MOV    $TPS,3$      ;SETUP ADDRESSES OF TP
4840 006176 016767 172740 000134  MOV    $TPS,5$      ;STATUS AND TP BUFFER
4841 006204 016767 172732 000140  MOV    $TPS,POINT1+2 ;INCASE THAY ARE NOT
4842 006212 016767 172724 000202  MOV    $TPS,$$TPS
4843 006220 016767 172720 000074  MOV    $TPB,2$      ;STANDARD ADDRESSES
4844 006226 016767 172712 000100  MOV    $TPB,4$
4845 006234 012706 001100  MOV    #STACK,SP    ;INITIALIZE THE SP
4846 006240 012705 001164  MOV    #TMP1,R5      ;PUT ADDRESS OF $TMP1 IN R5
4847 006244 012701 001166  MOV    #TMP2,R1      ;PUT ADDRESS OF $TMP2 IN R1
4848 006250 012702 100000  MOV    #BIT15,R2     ;SET SIGN BIT IN R2
4849 006254 010215  MOV    R2,(R5)       ;SET SIGN BIT IN $TMP1
4850 006256 005011  CLR    (R1)          ;CLEAR $TMP2
4851 006260 005002  CLR    R2            ;PUT ADDRESS OF LOCATION 0 IN R2
4852 006262 005012  CLR    (R2)          ;CLEAR LOCATION ZERO
4853 006264 012702 000064  MOV    #64,R2        ;PUT ADDRESS OF PRINTER VECTOR IN R2
4854 006270 012704 006350  MOV    #POINT1,R4    ;PUT ADDRESS OF POINT1 IN R4
4855 006274 010412  MOV    R4,(R2)       ;PUT ADDRESS OF POINT1 IN PRINTER VECTOR
4856 006276 012702 177776  MOV    #PSW,R2       ;PUT ADDRESS OF PSW IN R2
4857 006302 005767 172572  TST    $PASS        ;IS THIS PASS 1?
4858 006306 001015  BNE    SYNC62       ;BRANCH IF NO
4859 006310 012703 000101  MOV    #101,R3       ;PUT ASCII FOR "A" IN R3
4860 006314 012704 000100  MOV    #BIT06,R4     ;PUT INTERRUPT ENABLE BIT IN R4
4861 006320 010337  MOV    R3,$(PC)+    ;SEND A TO PRINTER
4862 006322 177566 2$:      .WORD 177566      ;ADDRESS OF TP BUFFER
4863 006324 105737 1$:      TSTB  $(PC)+    ;WAIT FOR PRINTER DONE
4864 006326 177564 3$:      .WORD 177564      ;INCASE DOUBLE BUFFERED
4865 006330 100375 1$:      BPL    1$          ;ADDRESS OF TP STATUS
    
```



```

4866 006332 010337          MOV      R3,@(PC)+      ;SEND SECOND A
4867 006334 177566          4$:      .WORD      177566
4868 006336 010437          MOV      R4,@(PC)+      ;SET THE INTERRUPT FLSG IN TPS
4869 006340 177564          5$:      .WORD      177564
4870 006342
4871 006342 000261          SYNC62:  SEC          ;SET C TO CATCH FAILURE TO ASH.20
4872 006344
4873 006344 012511          IUT62:   MOV      (R5)+,(R1) ;EXECUTE INSTRUCTION UNDER TEST
4874 006346 011202          MOV      (R2),R2        ;SAVE PSW IN R2
4875 006350 040437          POINT1: BIC      R4,@(PC)+ ;CLEAR THE INTERR FLAG
4876 006352 177564          .WORD      177564
4877 006354 005701          TST      R1              ;DID A DMO GET EXECUTED?
4878 006356 100001          BPL      3$              ;BRANCH IF NO
4879 006360 000000          HALT
4880
4881 006362 005711          3$:      TST      (R1)      ;IRCC DMO L STUCK LOW
4882 006364 100401          BMI      5$              ;FOR LOOPING CHANGE TO "BR TST62+2" (703)
4883 006366 000000          HALT                    ;DID A SM2*DM4 GET EXECUTED?
4884
4885 006370 011104          5$:      MOV      (R1),R4      ;BRANCH IF NO
4886 006372 020504          CMP      R5,R4          ;IRCC C FORK MUX INPUT B1 IS STUCK LOW
4887 006374 001001          BNE      6$              ;FOR LOOPING CHANGE TO "BR TST62+2" (700)
4888 006376 000000          HALT                    ;GET CONTENTS OF $TMP2
4889
4890 006400 022706 001074          6$:      CMP      #1074,SP    ;DID D12.01 GET EXECUTED?
4891 006404 001001          BNE      2$              ;BRANCH IF NO
4892 006406 000000          HALT                    ;RACE E29 IS BAD
4893
4894
4895 006410 005004          2$:      CLR      R4              ;FOR LOOPING CHANGE TO "BR TST62+2" (674)
4896 006412 005714          TST      (R4)          ;DID INSTRUCTION CAUSE WAIT TO OCCUR?
4897 006414 100001          BPL      IT              ;BRANCH IF NO
4898 006416 000000          HALT                    ;EITHER IRCC DSTMD H IS STUCK LOW OR
4899
4900 006420 105737          IT:      TSTB     @(PC)+  ;IT IS NOT GETTING THRU RACL RADROS
4901 006422 177564          $$TPS:  .WORD      177564  ;FOR LOOPING CHANGE TO "BR TST62+2" (670)
4902 006424 100375          BPL      IT              ;PUT ADDR. OF LOCATION ZERO IN R4
4903 006426 032702 000001          BIT      #BIT0,R2      ;DID A SM1*DM3 GET EXECUTED?
4904 006432 001001          BNE      TST63          ;BRANCH IF NO
4905 006434 000000          HALT                    ;IRCC C FORK MUX INPUT B2 IS STUCK LOW
4906
4907
4908
4909
4910
4911
4912
4913
4914
4915
4916
4917
4918
4919
4920
4921

```

```

;*****
;THE LOGICAL FLOW AT THIS POINT WOULD TEST A BIN*SM12*DM12*SRO(0)*DRO(0)
;*[TST.B+BIT.B+CMP.B] INSTRUCTION BUT NO ADDITIONAL LOGIC WOULD BE TESTED.
;*****

```

```

;*****
;TEST 63      FIVE MICROSTATES (BIN*SM12*DM12*SRO(1)*DRO(0)*CMPB)
;
;      THE ONLY THING THAT SHOULD FAIL WOULD BE STATE D12.90(110).
;
;      ROM FLOW-21,27,110,175,33
;*****

```

4922 006436 005200
4923 006440 012705 001164
4924 006444 112715 000377
4925 006450 012701 001166
4926 006454 012711 177400
4927 006450 005201
4928 006462 000240
4929 006464
4930 006464 121115
4931 006466 001401
4932 00647C 000000
4933
4934
4935
4936
4937
4938
4939
4940
4941
4942
4943
4944 006472 005200
4945 006474 012705 001164
4946 006500 010501
4947 006502 005025
4948 006504 012715 100000
4949 006510 000240
4950 006512
4951 006512 011545
4952 006514 022705 001170
4953 006520 001001
4954 006522 000000
4955
4956 006524 005711
4957 006526 100401
4958 006530 000000
4959
4960
4961
4962
4963
4964
4965
4966
4967
4968
4969
4970
4971
4972
4973
4974
4975
4976
4977

TST63: INC R0 ; INCREMENT TEST NUMBER
MOV #STMP1,R5 ; PUT ADDRESS OF STMP1 IN R5
MOVB #377,(R5) ; SET LOW BYTE OF STMP1 TO ALL ONE'S
MOV #STMP2,R1 ; PUT ADDRESS OF STMP2 IN R1
MOV #177400,(R1) ; SET HIGH BYTE OF STMP2 TO ALL ONES
INC R1 ; ADJUST R1 TO STMP2 HIGH BYTE
SYNC63: NOP
IUT63: CMPB (R1),(R5) ; EXECUTE INSTRUCTION UNDER TEST
BEQ TST64 ; TEST OK, GO TO NEXT TEST
HALT ; STATE D12.90 FAILED
; FOR LOOPING CHANGE TO "BR TST63+2" (763)

*TEST 64 FIVE MICROSTATES (BIN*SM12*DM12*0/CLASS)
*
* IF FORK C FAILS EXECUTION WILL GO TO D12.80
* WHICH WILL EXECUTE A SM1*DM2 TYPE INSTRUCTION.
*
* IF THE INSTRUCTION FAILS EITHER D45.80 OR D40.20 FAILED.
*
* RCM FLOW-21,27,115,121,157

TST64: INC R0 ; INCREMENT TEST NUMBER
MOV #STMP1,R5 ; PUT ADDRESS OF STMP1 IN R5
MOV R5,R1 ; SAVE ADDRESS OF STMP1
CLR (R5)+ ; CLEAR STMP1 AND STEP R5 TO STMP2
MOV #BIT15,(R5) ; SET SIGN BIT IN STMP2
SYNC64: NOP
IUT64: MOV (R5),-(R5) ; EXECUTE INSTRUCTION UNDER TEST
CMP #STMP2+2,R5 ; DID R5 AUTO INCREMENT?
BNE IS ; BRANCH IF YES
HALT ; IRCC FORK C MUX INPUT B1 STUCK HIGH
; FOR LOOPING CHANGE TO "BR TST64+2" (764)
IS: TST (R1) ; DID INSTRUCTION WORK?
BMI TST65 ; BRANCH IF YES
HALT ; EITHER STATE D45.80 OR D40.20 FAILED
; FOR LOOPING CHANGE TO "BR TST64+2" (761)
.SBTTL

*TEST 65 SIX MICROSTATES (DAC*DM12*ASRB*DR0(1))
*
* NEITHER FORK A NOR BEN15 NOR BEN05*FTN2 SHOULD FAIL
* SINCE THEY HAVE ALREADY BEEN TESTED.
*
* IF FORK B FAILS AFTER D12.30 EXECUTION WILL GO TO
* ONE OF THE FOLLOWING: RSD.00,D45.00,EXC.20,S45.00,
* CCP.00,MUL.00,SVC.10,MFP.00 OR DEP.00.
* RSD.00 WILL CAUSE A TRAP TO LOCATION 10.
* THIS WILL HAPPEN IF THE B FORK MUX SELECT IS STUCK LOW.
* IF STATE D45.00 IS ENTERED THE PROCESSOR WILL HANG UP
* IN A LOOP BETWEEN STATES D45.00 AND D10.30.
* IF STATE S45.00 IS ENTERED EXECUTION WILL GO TO STATE
* D12.80 AFTER S13.10 WHICH WILL HANG UP THE PROCESSOR.
* STATE CCP.00 WOULD SET OR CLEAR THE CONDITION CODES
* ACCORDING TO IR(4:0).

4978
4979
4980
4981
4982
4983
4984
4985
4986
4987
4988
4989
4990
4991
4992
4993
4994
4995
4996
4997
4998
4999
5000
5001
5002
5003
5004
5005
5006
5007
5008
5009
5010
5011
5012
5013
5014
5015
5016
5017
5018
5019
5020
5021
5022
5023
5024
5025
5026
5027
5028
5029
5030
5031
5032
5033

006532 005200
006534 012706 001074
006540 012705 001164
006544 012701 006572
006550 010115
006552 005205
006554 005002
006556 012703 000001
006562 012701 177776
006566 000264
006570
006570
006570 106215
006572 011102
006574 010567 172354
006600 005703
006602 001001
006604 000000
006606 012701 006572
006612 000301
006614 106001
006616 012703 001164
006622 020113
006624 001001
006626 000000
006630 022716 006572
006634 001001
006636 000000
006640 032702 000004
006644 001401
006646 000000
006650 012701 006572
006654 010105
006656 006001
006660 042701 000377
006664 042705 177400
006670 050501
006672 020167 172266

```

:* STATE MUL.00 WOULD CAUSE THE DESTINATION OPERAND TO
:* BE MULTIPLIED BY REGISTER 2 AND THE RESULT WOULD BE
:* STORED IN REGISTER 2 AND 3.
:* IF SVC.10 IS ENTERED A TRAP TO 4 WILL OCCUR BECAUSE
:* THE DESTINATION REGISTER IS ODD. THIS WILL ONLY HAPPEN
:* IF EITHER B FORK MUX INPUT B3 OR IRCB B0 RAB00 IS STUCK LOW.
:* IF MFP.00 IS ENTERED AN MFPI INSTRUCTION WILL BE EXECUTED
:* THIS WILL PUSH THE ADDRESS OF 1$ ONTO THE STACK.
:* DEP.00 WILL CAUSE THE PROCESSOR TO HANG IN MICRO ADDRESS
:* 170 WITH THE RUN LIGHT ON.
:*
:* ROM FLOW-1,175,137,64,123,132
:* *****
TST65: INC R0 ;INCREMENT TEST NUMBER
MOV #1074,SP ;INITIALIZE THE STACK
MOV #STMP1,R5 ;PUT ADDRESS OF STMP1 IN R5
MOV #AFTER,R1 ;PUT ADDRESS OF AFTER IN R1
MOV R1,(R5) ;STORE IN STMP1
INC R5 ;SET R5 TO HIGH BYTE OF STMP1
CLR R2 ;ENSURE R2 CLEAR
MOV #1,R3 ;ENSURE R3 NOT CLEAR
MOV #PSW,R1 ;PUT ADDRESS OF PSW IN R1
SEZ ;ENSURE Z BIT SET

SYNC65:
IUT65:
AFTER: ASRB (R5) ;EXECUTE INSTRUCTION UNDER TEST
MOV (R1),R2 ;SAVE PSW
MOV R5,$REGO ;SAVE R5
TST R3 ;DID MULTIPLY OCCUR & CLEAR R3?
BNE 3$ ;BRANCH IF NO
;B FORK MUX INPUT B1 STUCK HIGH
;FOR LOOPING CHANGE TO "BR TST65+2" (753)
MOV #AFTER,R1 ;PUT ADDRESS OF 1$ IN R1
SWAB R1 ;REVERSE BYTES
RORB R1 ;MAKE IT LOOK LIKE EXC.00 WAS ENTERED
MOV #STMP1,R3 ;PUT ADDRESS OF STMP1 IN R3
CMP R1,(R3) ;DID EXC.00 GET ENTERED?
BNE 4$ ;BRANCH IF NO
;IRCB 0B0(ASRB OR RORB) STUCK HIGH
;FOR LOOPING CHANGE TO "BR TST65+2" (742)
CMP #AFTER,(SP) ;DID MFP.00 EXECUTE?
BNE 5$ ;BRANCH IF NO
;B FORK MUX INPUT B2 STUCK LOW
;FOR LOOPING CHANGE TO "BR TST65+2" (736)
BIT #4,R2 ;DID CCP.00 EXECUTE?
BEQ 6$ ;BRANCH IF NO
;IRCC B0 RAB04 NOT GETTING THRU RA0L RADR54
;FOR LOOPING CHANGE TO "BR TST65+2" (732)
MOV #AFTER,R1 ;GET ADDRESS OF AFTER
MOV R1,R5 ;SAVE R1
ROR R1 ;RIGHT SHIFT R1 WITHOUT USING ASR
BIC #377,R1 ;CLEAR LOWER BYTE OF R1
BIC #177400,R5 ;CLEAR UPPER BYTE OF R5
BIS R5,R1 ;MAKE LOWER BYTE OF R10W
;BYTE OF DST. OPERAND
CMP R1,STMP1 ;DID DESTINATION GET ASR'D PROPERLY?

```

5034 006676 001401
5035 006700 000000

BEQ TST66
HALT

:: BRANCH IF YES
:: EITHER SHR.00 OR SHR.10 FAILED
:: FOR LOOPING CHANGE TO "BR TST65+2" (715)

5036
5037
5038
5039
5040
5041
5042
5043
5044
5045
5046
5047

:: *****
* TEST 66 SIX MICROSTATES (DAC*DM12*RORB*DRO(1))
*
* THIS TEST IS THE SAME AS THE LAST ONE EXCEPT A RORB
* IS USED INSTEAD OF AN ASRB.
*
* FORK B WILL ONLY FAIL IF IRCB E36(13) IS STUCK HIGH
* WHICH WILL CAUSE EXECUTION TO GO TO EXC.CO.
*
* ROM FLOW-2,175,137,64,123,132
* *****

5048 006702 005200
5049 006704 012705 001164
5050 006710 112725 000100
5051 006714 112715 000200
5052 006720 000240
5053 006722
5054 006722 106025
5055 006724 022745 040100
5056 006730 001401
5057 006732 000000

TST66: INC R0 ; INCREMENT TEST NUMBER
MOV #STMP1,R5 ; PUT ADDRESS OF STMP1 HIGH BYTE IN R5
MOV #100,(R5)+ ; SET BIT 6 IN LOW BYTE OF STMP1
MOV #200,(R5) ; SET SIGN BIT IN HIGH BYTE OF STMP1

SYNC66: NOP
IUT66: RORB (R5)+ ; EXECUTE INSTRUCTION UNDER TEST
CMP #40100,-(R5) ; DID INSTRUCTION WORK?
BEQ TST67 ; BRANCH IF YES
HALT ; IRCB E36(13) NOT GOING LOW ON RORB
; FOR LOOPING CHANGE TO "BR TST66+2" (764)

5058
5059
5060
5061
5062
5063
5064
5065
5066
5067
5068
5069
5070

:: *****
* THE LOGICAL FLOW AT THIS POINT WOULD TEST A DAC*DM3*[TST.B+BIT.B+CMPI.B]*DRO(0)
* FOLLOWED BY A DAC*DM4*P/CLASS*DRO(0) INSTRUCTION BUT NO ADDITIONAL LOGIC
* IS TESTED
* *****

5071
5072
5073
5074
5075
5076
5077
5078
5079

:: *****
* TEST 67 SIX MICROSTATES (DAC*DM6*XOR*DRO(0))
*
* IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TRAP TO 10.
* THIS SHOULD ONLY HAPPEN IF RACE E35(1) IS BAD.
*
* IF THE INSTRUCTION DOESN'T WORK IT WILL HALT IN THIS TEST.
*
* ROM FLOW-6,251,122,177,31,132
* *****

5080 006734 005200
5081 006736 005004
5082 006740 005104
5083 006742 010437 001164
5084 006746 000240
5085 006750
5086 006750 074467 172210
5087 006754 005767 172204
5088 006760 001401
5089 006762 000000

TST67: INC R0 ; INCREMENT TEST NUMBER
CLR R4 ; SETUP R4
COM R4 ; SET ALL BITS IN R4
MOV R4,#STMP1 ; SET ALL BITS IN STMP1

SYNC67: NOP
IUT67: XOR R4,STMP1 ; EXECUTE INSTRUCTION UNDER TEST
TST STMP1 ; DID STMP1 CLEAR?
BEQ TST70 ; BRANCH IF YES
HALT ; INSTRUCTION FAILED

;FOR LOOPING CHANGE TO "BR TST67+2" (765)

S090
S091
S092
S093
S094
S095
S096
S097
S098
S099
S100
S101
S102
S103
S104
S105
S106
S107
S108
S109
S110
S111
S112
S113
S114
S115
S116
S117
S118
S119
S120
S121
S122
S123
S124
S125
S126
S127
S128
S129
S130
S131
S132
S133
S134
S135
S136
S137
S138
S139
S140
S141
S142
S143
S144
S145

;THE LOGICAL SEQUENCE WOULD TEST A DAC*DM6*[TST.B+BIT.B+CMP.B]*DRO(1) BUT ALL
;THE LOGIC HAS BEEN TESTED.

;TEST 70 SIX MICROSTATES (NEG.B*DM12*DRO(0))

NEITHER FORK A NOR BGN15 SHOULD FAIL.

IF FORK B FAILS EXECUTION WILL GO TO RSD.00 CAUSING
A TRAP TO LOCATION 10 OR EXC.00.
RSD.00 SHOULD ONLY OCCUR IF THE B FORK MUX
STROBE IS BEING HELD LOW(CHIP FAILURE).

ROM FLOW-1,175,67,271,163,132

TST70: INC R0 ;INCREMENT TEST NUMBER
CLR STMP1 ;ENSURE STMP1 CLEAR
INC STMP1 ;PUT 1 IN STMP1
MOV #STMP1,R5 ;PUT ADDRESS OF STMP1 IN R5

SYNC70: NOP
IUT70: NEG (R5) ;EXECUTE INSTRUCTION UNDER TEST
CMP #177777,(R5) ;DID STMP1 NEGATE?
BEQ 35 ;BRANCH IF YES
CMP #177776,(R5) ;DID STMP1 COMPLEMENT?
BNE 15 ;BRANCH IF NO
HALT ;EITHER STATE NEG.10 FAILED
;OR FORK B FAILED. IPCB NEG.B H
;IS STUCK HIGH.

15: CMP #1,(R5) ;FOR LOOPING CHANGE TO "BR TST70+2" (761)
BNE 25 ;DID STMP1 STAY THE SAME?
HALT ;BRANCH IF NO
;STMP1 DID NOT GET LOADED
;FOR LOOPING CHANGE TO "BR TST70+2" (755)

25: HALT ;INSTRUCTION FAILED
;FOR LOOPING CHANGE TO "BR TST70+2" (754)

35: SEZ ;ENSURE Z SET
NEG (R5) ;EXECUTE INSTRUCTION UNDER TEST
BNE TST71 ;CC'S OK
HALT ;STATE NEG.10 BAD
;FOR LOOPING CHANGE TO "BR TST70+2" (750)

;THE LOGICAL SEQUENCE WOULD NEXT EXECUTE A JMP*DM3 BUT NO

5146
5147
5148
5149
5150
5151
5152
5153
5154
5155
5156
5157
5158
5159
5160
5161 007046 005200
5162 007050 012705 001164
5163 007054 012715 007064
5164 007060 000240
5165 007062
5166 007062 013501
5167 007064 026701 177774
5168 007070 001405
5169 007072 022701 007064
5170 007076 001001
5171 007100 000000
5172
5173
5174 007102
5175 007102 000000
5176
5177
5178
5179
5180
5181
5182
5183
5184
5185
5186
5187
5188
5189
5190
5191
5192
5193
5194
5195
5196
5197
5198
5199
5200 007104 005200
5201 007106 012705 100000

```

;*ADDITIONAL LOGIC IS TESTED.
;*****
;*****
;*****
;*TEST 71      SIX  MICROSTATES (BIN*SM3*DM0*-DF7*SRO(0))
;
;   FORK A SHOULD NOT FAIL.
;
;   IF BEN14*FEN4 FAILS EXECUTION WILL GO TO 000.90. THIS
;   WILL CAUSE A SM2 TO BE EXECUTED. THIS SHOULD ONLY
;   HAPPEN IF IRCC SM357 IS STUCK LOW OR NOT GETTING THRU RACL E70.
;
;   ROM FLOW-22,27,317,143,146,205
;*****
TST71:  INC      R0          ;INCREMENT TEST NUMBER
        MOV      #STMP1,R5  ;PUT ADDRESS OF STMP1 IN R5
        MOV      #POINT2,(R5) ;PUT ADDRESS OF POINT2 IN STMP1
SYNC71: NOP
IUT71:
POINT2: MOV      2(R5)+,R1   ;EXECUTE INSTRUCTION UNDER TEST
        CMP      POINT2,R1  ;DID R1 GET CORRECT DATA?
        BEQ      TST72      ;BRANCH IF YES
        CMP      #POINT2,R1 ;DID A SM2 GET EXECUTED?
        BNE      25         ;BRANCH IF NO
        HALT              ;EITHER IRCC SM357 STUCK LOW
                          ;OR NOT GETTING THRU RACL E70
                          ;FOR LOOPING CHANGE TO "BR TST71+2" (763)
25:     HALT              ;EITHER 513.20 OR 513.30 OR 513.40 FAILED
                          ;FOR LOOPING CHANGE TO "BR TST71+2" (762)
;*****
;THE LOGICAL SEQUENCE WOULD NEXT TEST A BIN*SM6*DM0*DF7*SRO(1), THEN A BIN*SM12*
;DM12*SRO(0)*DRO(1)*[TST.B+BIT.B+CMP.B] THEN A BIN*SM12*DM12*SRO(0)*
;DRO(0)*P/CLASS THEN A BIN*SM12*DM12*SRO(1)*DRO(1)*[TST.B+BIT.B+CMP.B]
;THEN A BIN*SM12*DM12*SRO(1)*DRO(0)*P/CLASS AND THEN A BIN*SM12*DM4*
;SRO(0)*DRO(0)*[TST.B+BIT.B+CMP.B] INSTRUCTION, BUT NO ADDITIONAL LOGIC
;IS TESTED.
;*****
;*****
;*****
;*TEST 72      SIX  MICROSTATES (BIN*SM12*DM4*SRO(1)*DRO(0)*CMPB)
;
;   A FAILURE WILL ONLY OCCUR IF STATE D45.90 FAILS.
;
;   IF THE DST REG. DOES NOT DECREMENT STATE D45.90 IS BAD.
;   IF THE CONDITION CODES ARE BAD THEN STATE D40.30
;   PROBABLY DID NOT SWAP THE BYTES OF THE SRC OPERAND.
;
;   ROM FLOW-1,27,114,131,177,33
;*****
TST72:  INC      R0          ;INCREMENT TEST NUMBER
        MOV      #BIT15,R5  ;SET SIGN BIT IN R5

```

```

5202 007112 010567 172046      MOV     R5,$TMP1      ;SET SIGN BIT IN $TMP1
5203 007116 012701 001166      MOV     #TMP2,R1     ;PUT ADDRESS OF $TMP2 IN R1
5204 007122 010105              MOV     R1,R5        ;PUT ADDRESS OF $TMP2 IN R5
5205 007124 112721 000200      MOVB   #BIT7,(R1)+   ;SET LOW BYTE SIGN IN $TMP2 & STEP R1
5206 007130 105011              CLRB   (R1)         ;ENSURE $TMP2 HIGH BYTE CLEAR
5207 007132 005305              DEC    R5           ;ADJUST R5 TO POINT AT $TMP1 HIGH BYTE
5208 007134 000240              SYNC72: NOP
5209 007136              IUT72:
5210 007136 121541              CMPB   (R5),-(R1)   ;EXECUTE INSTRUCTION UNDER TEST
5211 007140 0C1405              BEQ    TST73        ;TEST OK, GO TO NEXT TEST
5212 007142 020127 001166      CMP    R1,#TMP2     ;DID R1 DECREMENT?
5213 007146 001001              BNE   IS           ;BRANCH IF YES
5214 007150 000000              HALT                ;STATE D45.90 DID NOT DECREMENT
5215                                ;FOR LOOPING CHANGE TO "BR TST72+2" (756)
5216 007152              IS:
5217 007152 000000              HALT                ;INSTRUCTION FAILED
5218                                ;FOR LOOPING CHANGE TO "BR TST72+2" (755)

```

```

*****
;THE LOGICAL FLOW WOULD NEXT TEST A BIN*SM4*DM12*SRO(0)*DRO(0)*0/CLASS
;THEN A BIN*SM4*DM12*SRO(0)*DRO(0)*[TST.B+BIT.B+CMP.B] THEN A BIN*SM4*
;DM12*SRO(1)*DRO(0)*[TST.B+BIT.B+CMP.B] THEN A BIN*SM4*DM4*SRO(0)*DRO(0)*
;0/CLASS INSTRUCTION, BUT NO ADDITIONAL LOGIC IS TESTED.
*****

```

```

*****
;TEST 73      SIX  MICROSTATES (DAC*DM5*DRO(0)*0/CLASS)
;
;   FORK A SHOULD NOT FAIL.
;
;   IF IRCD DM357 IS STUCK LOW OR NOT GETTING THRU
;   TO RACK E51 A DM4 WILL BE EXECUTED.
;   IF STATE D10.00 OR D10.10 FAIL TO FETCH THE DEFERED ADDRESS
;   THE SOURCE WILL BE STORED IN THE DESTINATION.
;
;   ROM FLOW-5,162,231,233,311,157
*****

```

```

5241 007154 005200              TST73: INC    R0      ;INCREMENT TEST NUMBER
5242 007156 012705 100000      MOV     #BIT15,R5   ;SET SIGN BIT IN R5
5243 007162 012701 001166      MOV     #TMP2,R1     ;PUT ADDRESS OF $TMP2 IN R1
5244 007166 010167 171772      MOV     R1,$TMP1    ;PUT ADDRESS OF $TMP2 IN $TMP1
5245 007172 005011              CLR    (R1)         ;ENSURE $TMP2 CLEAR
5246 007174 000240              SYNC73: NOP
5247 007176              IUT73:
5248 007176 010551              MOV     R5,2-(R1)   ;EXECUTE INSTRUCTION UNDER TEST
5249 007200 005767 171762      TST    $TMP2        ;DID SIGN BIT GET SET IN $TMP2?
5250 007204 100405              BMI   TST74        ;BRANCJ IF YES
5251 007206 020567 171752      CMP    R5,$TMP1    ;DID MODE 4 GET EXECUTED?
5252 007212 001001              BNE   IS           ;EITHER IRCD DM357 IS NOT GOING LOW
5253 007214 000000              HALT                ;OR NOT GETTING THRU TO RACK E51
5254                                ;FOR LOOPING CHANGE TO "BR TST73+2" (760)
5255                                IS:
5256 007216              IS:
5257 007216 000000              HALT                ;INSTRUCTION FAILED

```

;FOR LOOPING CHANGE TO "BR TST73+2" (757)

5258
5259
5260
5261
5262
5263
5264
5265
5266
5267
5268
5269
5270
5271
5272
5273
5274
5275
5276
5277
5278
5279
5280
5281
5282
5283
5284
5285
5286
5287
5288
5289
5290
5291
5292
5293
5294
5295
5296
5297
5298
5299
5300
5301
5302
5303
5304
5305
5306
5307
5308
5309
5310
5311
5312
5313

*THE LOGICAL SEQUENCE WOULD NEXT TEST A DAC*DM3*DRO(0)*P/CLASS THEN A
*DAC*DM3*DRO(1)*[TST.B+BIT.B+CMP.B] THEN A DAC*DM4*DRO(1)*[ASRB+
*RORB] THEN A DAC*DM5*DRO(0)*[TST.B+BIT.B+CMP.B] THEN A DAC*DM6*
*DRO(1)*P/CLASS INSTRUCTION, BUT NO ADDITIONAL LOGIC IS TESTED.

.SBTTL

*TEST 74 SEVEN MICROSTATES (DAC*DM7*0/CLASS)

* FORK A SHOULD NOT FAIL.

* IF IRCD DM357 DOES NOT GO HIGH A DM6 WILL BE EXECUTED.

* ALL OTHER LOGIC HAS BEEN TESTED.

* ROM FLOW-7,251,162,231,233,311,157

TST74: INC R0 ;INCREMENT TEST NUMBER
MOV #STMP2,R5 ;PUT ADDRESS OF STMP2 IN R5
MOV R5,STMP1 ;PUT ADDRESS OF STMP2 IN STMP1
MOV #STMP0,R1 ;PUT ADDRESS OF STMP0 IN R1
MOV #BIT15,R2 ;SET SIGN BIT IN R2
CLR STMP2 ;ENSURE STMP2 CLEAR

SYNC74: NOP

IUT74:

MOV R2,02(R1) ;EXECUTE INSTRUCTION UNDER TEST
TST STMP2 ;DID STMP2 GET SIGN BIT SET?
BMI TST75 ;BRANCH IF YES
CMP R2,STMP1 ;DID DM6 GET EXECUTED?
BNE IS ;BRANCH IF NO
HALT ;IRCD DM357 DID NOT GO HIGH
;FOR LOOPING CHANGE TO "BR TST74+2" (754)

IS:

HALT ;INSTRUCTION FAILED
;FOR LOOPING CHANGE TO "BR TST74+2" (753)

*THE LOGICAL SEQUENCE WOULD NEXT TEST A NEG.B*DM12*DRO(1) THEN A NEG.B*DM4*DRO(0)
*THEN A JMP*DM5 INSTRUCTION, BUT NO ADDITIONAL LOGIC IS TESTED.

*TEST 75 SEVEN MICROSTATES (JSR*DM12)

* IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TRAP TO LOCATION 10.
* THIS WILL ONLY OCCUR IF RACE JMP+JSR+SWAB DOES NOT GO HIGH.

* IF EITHER IRCB IR(14:9)04 DOES NOT GO LOW OR E63 IS BAD


```

5314
5315
5316
5317
5318
5319
5320
5321
5322
5323 007274 005200
5324 007276 012706 001076
5325 007302 012705 007322
5326 007306 010701
5327 007310
5328 007310 000277
5329 007312
5330 007312 004125
5331 007314 100001
5332 007316 000000
5333
5334 007320
5335 007320 000000
5336
5337 007322 022716 007310
5338 007326 001401
5339 007330 000000
5340
5341 007332 022701 007314
5342 007336 001401
5343 007340 000000
5344
5345 007342 022706 001074
5346 007346 001401
5347 007350 000000
5348
5349
5350
5351
5352
5353
5354
5355
5356
5357
5358
5359
5360
5361
5362
5363
5364
5365
5366
5367
5368
5369 007352 005200

```

```

;* EXECUTION WILL GO FROM D12.10 TO EXC.00.
;* IF IRCB E63 IS BAD (PIN 10 OR 4&5 FLOATING) EXECUTION WILL
;* GO TO RSD.00 CAUSING A TRAP TO LOCATION 10. THIS FAILURE
;* WOULD INCREMENT THE DST REG. BEFORE THE TRAP.
;*
;* IF THE INSTRUCTION FAILS THEN ONE OF THE JSR STATES FAILED.
;*
;* ROM FLOW-2,135,34,201,274,275,32
;*****
TST75: INC RO ;INC TST NUMBER
MOV #1076,SP ;INITIALIZE SP
MOV #T67,R5 ;PUT ADDRESS OF T67A IN R5
MOV PC,R1 ;PUT RANDOM NUMBER IN R1

SYNC75:
T67A: SCC ;ENSURE ALL CC'S SET
IUT75:
T67B: JSR R1,(R5)+ ;EXECUTE INSTRUCTION UNDER TEST
BPL T67C ;BRANCH IF N CLEARED
HALT ;PCB DID NOT LOAD
;FOR LOOPING CHANGE TO "BR TST75+2" (767)

T67C: HALT ;FORK B FAILED TO EXC.00(SEE ABOVE)
;FOR LOOPING CHANGE TO "BR TST75+2" (766)

T67: CMP #T67A,(SP) ;DID R1 GET STACKED?
BEQ 4$ ;BRANCH IF YES
HALT ;REGISTER DID NOT GET STACKED
;FOR LOOPING CHANGE TO "BR TST75+2" (762)

4$: CMP #T67B,R1 ;DID R1 GET LOADED?
BEQ 5$ ;BRANCH IF YES
HALT ;REGISTER DID NOT LOAD
;FOR LOOPING CHANGE TO "BR TST75+2" (756)

5$: CMP #1074,SP ;DID SP GET DECREMENTED?
BEQ TST76 ;BRANCH IF YES
HALT ;SP DID NOT DECREMENT
;FOR LOOPING CHANGE TO "BR TST75+2" (752)

;*****
;THE LOGICAL SEQUENCE WOULD NEXT EXECUTE A BIN*SM3*DM0*-DF7*SRO(1) THEN
;A BIN*SM3*DM0*DF7*SRO(0) THEN A BIN*SM3*DM0*DF7*SRO(1) INSTRUCTION,
;BUT NO ADDITIONAL LOGIC IS TESTED.
;*****

;*****
;TEST 76 SEVEN MICROSTATES (BIN*SM5*DM0*-DF7*SRO(0))
;*
;* FORK A SHOULD NOT FAIL.
;*
;* IF BEN14*FEN4 FAILS EXECUTION WILL GO TO D00.90
;* CAUSING A SM4 INSTRUCTION TO BE EXECUTED. THIS WILL ONLY
;* OCCUR IF EITHER IRCC SRCMS DOES NOT GO LOW OR IF IRCC E28 IS BAD.
;*
;* ROM FLOW-24,23,27,317,143,146,205
;*****
TST76: INC RO ;INCREMENT TEST NUMBER

```

```

5370 007354 012705 001166      MOV      #STMP2,R5      ;PUT ADDRESS OF STMP2 IN R5
5371 007360 010567 171600      MOV      R5,STMP1     ;PUT ADDRESS OF STMP2 IN STMP1
5372 007364 012715 100000      MOV      #BIT15,(R5)  ;SET SIGN BIT IN STMP2
5373 007370 005001              CLR      R1            ;ENSURE R1 CLEAR
5374 007372 000240      SYNC76: NOP
5375 007374              IUT76:
5376 007374 015501              MOV      2-(R5),R1    ;EXECUTE INSTRUCTION UNDER TEST
5377 007376 022701 100000      CMP      #BIT15,R1    ;DID INSTRUCTION WORK?
5378 007402 001405              BEQ      TST77        ;BRANCH IF YES
5379 007404 020167 171554      CMP      R1,STMP1    ;DID MODE 4 EXECUTE?
5380 007410 001001              BNE     1$           ;BRANCH IF NO
5381 007412 000000              HALT                ;EITHER IRCC SRCMS NOT GOING LOW OR IRCC E28 BAD
5382                                ;FOR LOOPING CHANGE TO "BR TST76+2" (760)
5383 007414              1$:
5384 007414 000000              HALT                ;INSTRUCTION FAILED
5385                                ;FOR LOOPING CHANGE TO "BR TST76+2" (757)
5386
5387      ;*****
5388      ;*THE LOGICAL SEQUENCE WOULD NEXT EXECUTE A BIN*SM12*DM12*SRO(0)*DRO(1)*
5389      ;*P/CLASS THEN A BIN*SM12*DM12*SRO(1)*DRO(1)P/CLASS INSTRUCTION, BUT
5390      ;*NO ADDITIONAL LOGIC IS TESTED.
5391      ;*****
5392
5393
5394      ;*****
5395      ;*TEST 77 SEVEN MICROSTATES (BIN*+SM12*DM3*0/CLASS)
5396      ;*
5397      ;* IF IRCC C FORK MUX INPUT B2 IS NOT GOING LOW OR IRCC E40
5398      ;* IS BAD A DM2 WILL BE EXECUTED.
5399      ;*
5400      ;* THE ONLY OTHER POSSIBLE FAILURE IS STATE D30.80.
5401      ;*
5402      ;* ROM FLOW-21,27,113,221,233,311,157
5403      ;*****
5404 007416 005200      TST77: INC      R0      ;INCREMENT TEST NUMBER
5405 007420 005067 171542      CLR      STMP2        ;ENSURE STMP2 CLEAR
5406 007424 012705 001164      MOV      #STMP1,R5    ;PUT ADDRESS OF STMP1 IN R5
5407 007430 012715 001166      MOV      #STMP2,(R5)  ;PUT ADDRESS OF STMP2 IN STMP1
5408 007434 000240      SYNC77: NOP
5409 007436              IUT77:
5410 007436 011535      MOV      (R5),2(R5)+  ;EXECUTE INSTRUCTION UNDER TEST
5411 007440 024515      CMP      -(R5),(R5)   ;DID INSTRUCTION WORK?
5412 007442 001405      BEQ      TST100       ;BRANCH IF YES
5413 007444 022745 001166      CMP      #STMP2,-(R5) ;DID DM2 GET EXECUTED?
5414 007450 001001      BNE     1$           ;BRANCH IF NO
5415 007452 000000      HALT                ;EITHER C FORK MUX INPUT B2
5416                                ;NOT GOING LOW OR IRCC E40 BAD
5417                                ;FOR LOOPING CHANGE TO "BR TST77+2" (762)
5418 007454              1$:
5419 007454 000000      HALT                ;INSTRUCTION FAILED
5420                                ;FOR LOOPING CHANGE TO "BR TST77+2" (761)
5421
5422
5423      ;*****
5424      ;*THE LOGICAL SEQUENCE WOULD NEXT TEST A BIN*SM12*DM4*SRO(0)*DRO(0)*
5425      ;*P/CLASS FOLLOWED BY A BIN*SM12*DM4*SRO(0)*DRO(1)*[TST.B+BIT.B+CMP.B]

```

K09

5426
5427
5428
5429
5430
5431
5432
5433
5434
5435
5436
5437
5438
5439
5440
5441
5442
5443
5444
5445
5446
5447
5448
5449
5450
5451
5452
5453
5454
5455
5456
5457
5458
5459
5460
5461
5462
5463
5464
5465
5466
5467
5468
5469
5470
5471
5472
5473
5474
5475
5476
5477
5478
5479
5480
5481

007456 005200
007460 012706 001076
007464 012746 000340
007470 012746 007526
007474 005067 171466
007500 012705 001164
007504 012715 100000
007510 000240
007512
007512 011565 000002
007516 005767 171444
007522 100402
007524 000000
007526
007526 000000

; *FOLLOWED BY A BIN*SM12*DM4*SRO(1)*DRO(0)*P/CLASS FOLLOWED BY A
; *BIN*SM12*DM4*SRO(1)*DRO(1)*[TST.B+BIT.B+CMP.B] INSTRUCTION, BUT NO
; *ADDITIONAL LOGIC IS TESTED.
; *****
; *****
; *TEST 100 SEVEN MICROSTATES (BIN*SM12*DM6*0/CLASS)
; *
; * IF FORK C FAILS EXECUTION WILL GO TO D45.90 AND A
; * DM4 WILL BE EXECUTED. THIS WILL ONLY HAPPEN IF IRCC E39 PIN 5
; * IS NOT GOING LOW. THIS WILL CAUSE AN RTI SINCE THE LOCATION
; * FOLLOWING THE INSTRUCTION CONTAINS 000002.
; *
; * THE ONLY OTHER FAILURE WOULD BE CAUSED BY STATE D67.80 BEING BAD.
; *
; * ROM FLOW-21,27,117,6,251,122,157
; * *****
TST100: INC R0 ; INCREMENT TEST NUMBER
MOV #1076,SP ; INITIALIZE THE SP
MOV #PR7,-(SP) ; PUT PRIORITY LEVEL 7 ON STACK
MOV #T73,-(SP) ; PUT ADDRESS OF T73 ON STACK
CLR \$TMP2 ; ENSURE \$TMP2 CLEAR
MOV \$TMP1,R5 ; PUT ADDRESS OF \$TMP1 IN R5
MOV #BIT15,(R5) ; SET SIGN BIT IN \$TMP1
SYN100: NOP
IUT100: MOV (R5),2(R5) ; EXECUTE INSTRUCTION UNDER TEST
TST \$TMP2 ; DID INSTRUCTION WORK?
BMI TST101 ; BRANCH IF YES
HALT ; INSTRUCTION FAILED
; FOR LOOPING CHANGE TO "BR TST100+2" (755)
T73: HALT ; IRCC E39(5) IS NOT GOING LOW
; FOR LOOPING CHANGE TO "BR TST100+2" (754)

; *****
; *THE LOGICAL SEQUENCE WOULD NEXT TEST THE INSTRUCTIONS BETWEEN
; *BIN*SM4*DM12*SRO(0)*DRO(1)*[TST.B+BIT.B+CMP.B] AND BIN*SM5*DM0*DF7*SRO(1)
; *BUT NOT ADDITIONAL LOGIC IS TESTED.
; *****

.SB7TL
; *****
; *TEST 101 EIGHT MICROSTATES (BIN*SM7*DM0*-DF7*SRO(0))
; *
; * FORK A SHOULD NOT FAIL.
; *
; * IF FEN4*BEN14 FAILS EXECUTION WILL GO TO D00.90 CAUSING
; * A SM6 TO BE EXECUTED. THIS WILL ONLY HAPPEN IF EITHER
; * IRCC SRCM7 DOES NOT GO LOW OR IF IRCC E28(1) IS BAD.
; *
; * ROM FLOW-26,54,141,142,317,143,146,205
; * *****

```

5482 007530 005200          TST101: INC      R0          ;INCREMENT TEST NUMBER
5483 007532 012767 001164 171426      MOV      #STMP1,$TMP2 ;PUT ADDRESS OF STMP1 IN $TMP2
5484 007540 012767 100000 171416      MOV      #BIT15,$TMP1 ;SET SIGN BIT IN $TMP1
5485 007546 012705 001164          MOV      #STMP1,R5    ;PUT ADDRESS OF $TMP1 IN R5
5486 007552 005001          CLR      R1          ;ENSURE R1 CLEAR
5487 007554 000240          SYN101: NOP
5488 007556          IUT101:
5489 007556 017501 000002      MOV      @2(R5),R1    ;EXECUTE INSTRUCTION UNDER TEST
5490 007562 005701          TST      R1          ;DID R1 GET SIGN BIT SET?
5491 007564 100405          BMI      TST102     ;BRANCH IF YES
5492 007566 022701 001164      CMP      #STMP1,R1   ;DID SRCM6 GET EXECUTED?
5493 007572 001001          BNE      1$         ;BRANCH IF NO
5494 007574 000000          HALT             ;EITHER IRCC SRCM7 DOES NOT GO LOW OR IRCC E28 BAD
5495                                     ;FOR LOOPING CHANGE TO "BR TST101+2" (756)
5496 007576          1$:
5497 007576 000000          HALT             ;INSTRUCTION FAILED
5498                                     ;FOR LOOPING CHANGE TO "BR TST101+2" (755)
5499
5500
5501
5502
5503
5504
5505
5506
5507
5508
5509
5510
5511
5512
5513
5514
5515
5516
5517
5518
5519
5520
5521
5522
5523
5524
5525
5526
5527
5528
5529
5530
5531
5532
5533
5534
5535
5536
5537

```

```

;*****
;THE LOGICAL SEQUENCE WOULD NEXT TEST A BIN*SM12*DM3*SRO(0)*DRO(0)*[TST.B+
;BIT.B+CMP.B] BUT NO ADDITIONAL LOGIC IS TESTED.
;*****

```

```

;*****
;TEST 102      EIGHT MICROSTATES (BIN*SM12*DM3*SRO(1)*DRO(0)*CMPB)
;
;THE ONLY POSSIBLE FAILURE WOULD BE IN STATE D30.90
;SINCE ALL THE OTHER LOGIC HAS BEEN TESTED.
;
;ROM FLOW-21,27,112,221,233,311,177,33
;*****

```

```

5515 007600 005200          TST102: INC      R0          ;INCREMENT TEST NUMBER
5516 007602 012767 001166 171354      MOV      #STMP2,$TMP1 ;PUT ADDRESS OF $TMP2 IN $TMP1
5517 007610 012767 000377 171350      MOV      #377,$TMP2  ;SET LOW BYTE OF $TMP2 TO ALL ONES
5518 007616 012767 177400 171336      MOV      #177400,$TMP0 ;SET HIGH BYTE OF $TMP0 TO ALL ONES
5519 007624 012705 001163          MOV      #STMP0+1,R5  ;PUT ADDRESS OF $TMP0 HIGH BYTE IN R5
5520 007630 012701 001164          MOV      #STMP1,R1   ;PUT ADDRESS OF $TMP1 IN R1
5521 007634 000240          SYN102: NOP
5522 007636          IUT102:
5523 007636 121531          CMPB     (R5),@2(R1)+ ;EXECUTE INSTRUCTION UNDER TEST
5524 007640 001401          BEQ      TST103     ;BRANCH IF TEST OK
5525 007642 000000          HALT             ;STATE D30.90 FAILED
5526                                     ;FOR LOOPING CHANGE TO "BR TST102+2" (757)
5527
5528
5529
5530
5531
5532
5533
5534
5535
5536
5537

```

```

;*****
;THE LOGICAL SEQUENCE WOULD NEXT TEST INSTRUCTIONS BIN*SM12*DM4*SRO(0)*DRO(1)*
;P/CLASS THRU BIN*SM12*DM6*SRO(0)*DRO(0)*[TST.B+BIT.B+CMP.B] BUT NO
;ADDITIONAL LOGIC IS TESTED.
;*****

```

```

;*****
;TEST 103      EIGHT MICROSTATES (BIN*SM12*DM6*SRO(1)*DRO(0)*CMPB)

```

M09

```

5538
5539
5540
5541
5542
5543 007644 005200
5544 007646 012767 177400 171310
5545 007654 012705 001165
5546 007660 012767 000377 171300
5547 007666 000240
5548 007670
5549 007670 121567 171272
5550 007674 001401
5551 007676 000000
5552
5553
5554
5555
5556
5557
5558
5559
5560
5561 007700 005200
5562 007702 012705 001162
5563 007706 012767 000377 171246
5564 007714 012701 001166
5565 007720 012767 000377 171240
5566 007726 012767 001166 171230
5567 007734 000240
5568 007736
5569 007736 021551
5570 007740 001401
5571 007742 000000
5572
5573
5574
5575
5576
5577
5578 007744 005200
5579 007746 012705 001163
5580 007752 012767 177400 171202
5581 007760 012701 001166
5582 007764 012767 000377 171174
5583 007772 012767 001166 171164
5584 010000 121551
5585 010002 001401
5586 010004 000000
5587
5588
5589
5590
5591
5592
5593

```

```

;*
;* THE ONLY POSSIBLE FAILURE IN THIS TEST IS STATE D67.90.
;*
;* ROM FLOW-21,27,116,6,251,122,177,33
;* *****
TST103: INC RO ; INCREMENT TEST NUMBER
MOV #177400,$TMP1 ; SET HIGH BYTE OF $TMP1 TO ALL ONES
MOV #$TMP1+1,R5 ; PUT ADDRESS OF $TMP1 HI BYTE IN R5
MOV #377,$TMP2 ; SET LOW BYTE OF $TMP2 TO ALL ONES
SYN103: NOP
IUT103:
CMPB (R5),$TMP2 ; EXECUTE INSTRUCTION UNDER TEST
BEQ TST104 ; BRANCH IF TEST OK
HALT ; STATE D67.90 FAILED
; FOR LOOPING CHANGE TO "BR TST103+2" (763)
.SBTTL
; *****
; TEST 104 NINE MICROSTATES (BIN*SM12*DM5*SRO(0)*DRO(0)*CMP.B)
;*
;* THE ONLY POSSIBLE FAILURE IN THIS TEST IS STATE D50.20.
;*
;* ROM FLOW-21,27,115,161,231,233,311,177,33
;* *****
TST104: INC RO ; INCREMENT THE TEST NUMBER
MOV #$TMP0,R5 ; PUT ADDRESS OF $TMP0 IN R5
MOV #377,$TMP0 ; SET LOW BYTE OF $TMP0 TO ALL ONES
MOV #$TMP2,R1 ; PUT ADDRESS OF $TMP2 IN R1
MOV #377,$TMP2 ; SET LOW BYTE OF $TMP2 TO ALL ONES
MOV #$TMP2,$TMP1 ; PUT ADDRESS OF $TMP2 IN $TMP1
SYN104: NOP
IUT104:
CMPB (R5),2-(R1) ; EXECUTE INSTRUCTION UNDER TEST
BEQ TST105 ; GO TO NEXT TEST
HALT ; STATE D50.20 IS BAD
; FOR LOOPING CHANGE TO "BR TST104+2" (757)
; *****
; TEST 105 EIGHT MICROSTATES (BIN*SM12*DM5*SRO(1)*DRO(0)*CMPB)
;*
;* THE ONLY POSSIBLE FAILURE IN THIS TEST IS STATE D50.30.
;* *****
TST105: INC RO ; INCREMENT THE TEST NUMBER
MOV #$TMP0+1,R5 ; PUT ADDRESS OF $TMP0 HIGH BYTE IN R5
MOV #177400,$TMP0 ; SET HIGH BYTE OF $TMP0 TO ALL ONES
MOV #$TMP2,R1 ; PUT ADDRESS OF $TMP2 IN R1
MOV #377,$TMP2 ; SET LOW BYTE OF $TMP2 TO ALL ONES
MOV #$TMP2,$TMP1 ; PUT ADDRESS OF $TMP2 IN $TMP1
CMPB (R5),2-(R1) ; EXECUTE INSTRUCTION UNDER TEST
BEQ TST106 ; GO TO NEXT TEST
HALT ; STATE D50.30 IS BAD
; FOR LOOPING CHANGE TO "BR TST105+2" (760)
; *****
; TEST 106 WRITE/READ PSW
;*
;* THIS TEST VERIFIES THAT THE PSW CAN BE READ THRU THE DATA MUX.
;* IF THE TEST FAILS ONE OF MANY THINGS COULD BE BAD WHICH CANNOT BE

```

```

5594          ;*      DETERMINED IN THIS DIAGNOSTIC.
5595          ;*      THIS TEST REQUIRES THAT SCCE PS ADRS GETS TO TMC
5596          ;*      THAT THE TMC DMUX SELECT LINES GET TO PDR, THAT THE PSW
5597          ;*      BITS GET TO THE DMUX, THAT SCCE INTERNAL ADDRESS GETS TO TMC,
5598          ;*      AND SCCA VA00 GETS TO UBC.
5599          ;*      *****
5600 010006 005200          ;TST106: INC      RO          ;INCREMENT THE TEST NUMBER
5601 010010 012737 000240 177776      MOV      #PR5,2#PSW      ;SET PRIORITY BITS WITH A DATO
5602 010016 012701 000257              MOV      #257,R1        ;PUT VALUE OF CC'S IN R1
5603 010022 000277              SCC          ;SET ALL THE CC'S WITH A CCOP INSTR
5604 010024 020137 177776      CMP      R1,2#PSW      ;EXECUTE TEST MODE
5605 010030 001416              BEQ      TST107        ;BRANCH IF READ PSW WORKS
5606 010032 012701 177776      MOV      #PSW,R1        ;GET ADDRESS OF PSW
5607 010036 000277              SCC          ;SET ALL THE CONDITION CODES
5608 010040 020137 177776      CMP      R1,2#PSW      ;DID DMUX SELECT BUS REG?
5609 010044 001001              BNE      1$          ;BRANCH IF NO
5610 010046 000000              HALT          ;EITHER TMCB E29 BAD OR SCCE PS ADDRESS
5611          ;NOT GETTING TO TMCD AS A HIGH
5612          ;FOR LOOPING CHANGE TO "BR TST106+2" (760)
5613 010050 005001          1$:      CLR      R1
5614 010052 000277              SCC
5615 010054 020137 177776      CMP      R1,2#PSW      ;DOES TMCD LOW BYTE ENABLE GO HIGH?
5616 010060 001001              BNE      2$          ;BRANCH IF YES
5617 010062 000000              HALT          ;EITHER TMCD LO BYTE EN DOES
5618          ;NOT GO HIGH OR IT DOES NOT GET
5619          ;THRU TO PDRE
5620          ;FOR LOOPING CHANGE TO "BR TST106+2" (752)
5621 010064          2$:      HALT
5622 010064 000000          ;TEST FAILED, SEE TEST DESCRIPTION
5623          ;FOR LOOPING CHANGE TO "BR TST106+2" (751)
5624          ;*****
5625          ;TEST 107      RTI
5626          ;*
5627          ;*      IF FORK A FAILS EXECUTION WILL GO TO ONE OF THREE STATES.
5628          ;*      RSD.00 WILL CAUSE A TRAP TO LOCATION 4. THIS WOULD HAPPEN
5629          ;*      IF RACF (HALT:OP CD 7) DOES NOT GO HIGH.
5630          ;*      STATE D12.01 WOULD CAUSE AN ODD ADDRESS TRAP SINCE RO
5631          ;*      WILL CONTAIN A 1. THIS WILL HAPPEN IF RACE E7 IS BAD.
5632          ;*      HLT.00 WILL CAUSE THE PROCESSOR TO HALT ON THE INSTRUCTION
5633          ;*      UNDER TEST AND WILL OCCUR IF RACF E17 IS BAD.
5634          ;*      IF THE INSTRUCTION DOESN'T WORK THEN ONE OF THE RTI MACHINE
5635          ;*      STATES IS BAD.
5636          ;*
5637          ;*      ROM FLOW-12,156,212,213,214,215,172
5638          ;*      *****
5639 010066 005200          ;TST107: INC      RO          ;INCREMENT TEST NUMBER
5640 010070 012706 001100      MOV      #STACK,SP      ;INITIALIZE THE STACK
5641 010074 012746 000340      MOV      #PR7,-(SP)      ;PUT PRIORITY LEVEL 7 ON STACK
5642 010100 012746 010120      MOV      #T71,-(SP)      ;PUT ADDRESS OF T71 ON STACK
5643 010104 012705 000001      MOV      #1,R5          ;PUT ODD ADDRESS IN R5.
5644 010110 000240          SYN107: NOP
5645 010112          IUT107:
5646 010112 000002          RTI          ;EXECUTE INSTRUCTION UNDER TEST
5647 010114 000240          NOP          ;IF THE PROCESSOR HALTS HERE
5648          ;RACF E17 IS BAD(AFIR51(1))*(HALT:OP CD 7))
5649 010116 000000          HALT          ;PCB DID NOT GET LOADED

```



```

5706                                     ;DO NOT WORK
5707                                     ;FOR LOOPING CHANGE TO "BR TST110+2" (717)
5708 010304 4S: HALT
5709 010304 000000                                     ;EITHER IRCD EMT IS NOT GOING HIGH
5710                                     ;OR DAPE TV03 IS NOT GOING HIGH
5711                                     ;OR NOT GETTING TO THE ALU
5712                                     ;FOR LOOPING CHANGE TO "BR TST110+2" (716)
5713 010306 5S: HALT
5714 010306 000000                                     ;EITHER DAPE TV02 IS NOT GOING HIGH
5715                                     ;OR IT IS NOT GETTING TO THE ALU
5716                                     ;FOR LOOPING CHANGE TO "BR TST110+2" (715)
5717 010310 6S: HALT
5718 010310 000000                                     ;EITHER DAPE TV01 IS STUCK HIGH
5719                                     ;OR THE LOW IS NOT GETTING TO THE ALU
5720                                     ;FOR LOOPING CHANGE TO "BR TST110+2" (714)
5721 010312 7S: HALT
5722 010312 000000                                     ;EITHER DAPE TV04 IS STUCK HIGH OR
5723                                     ;THE LOW IS NOT GETTING TO THE ALU
5724                                     ;FOR LOOPING CHANGE TO "BR TST110+2" (713)
5725 010314 8S: HALT
5726 010314 000000                                     ;DAPE E24(B0) STUCK HIGH (CHIP FAILURE)
5727                                     ;FOR LOOPING CHANGE TO "BR TST110+2" (712)
5728 010316 9S: HALT
5729 010316 000000                                     ;DAPE E25(B0) STUCK HIGH(CHIP FAILURE)
5730                                     ;FOR LOOPING CHANGE TO "BR TST110+2" (711)
5731 010320 10S: HALT
5732 010320 000000                                     ;DAPE TV05*07 STUCK HIGH
5733                                     ;FOR LOOPING CHANGE TO "BR TST110+2" (710)
5734 ;NOTE: THE ONLY WAY THE TRAP INSTRUCTION SHOULD FAIL IS THAT IT GETS
5735 ;THE WRONG VECTOR. IF THIS HAPPENS A HALT IN LOW CORE SHOULD OCCUR.
5736 010322 012737 010352 000034 MOV #115,2#TRAPVEC ;PUT ADDRESS OF 115 IN TRAP VECTOR
5737 010330 012737 010346 000010 MOV #125,2#RESVEC ;SETUP RESVEC
5738 010336 012737 010350 000014 MOV #135,2#BPTVEC ;SETUP LOCATION 14
5739 010344 104777 TRAP 377 ;EXECUTE INSTRUCTION UNDER TEST
5740 010346 12S: HALT
5741 010346 000000                                     ;EITHER IRCD TRAP DOES NOT GO LOW
5742                                     ;OR IT IS NOT GETTING THRU DAPE
5743                                     ;FOR LOOPING CHANGE TO "BR TST110+2" (675)
5744 010350 13S: HALT
5745 010350 000000                                     ;DAPE E7 IS BAD
5746                                     ;FOR LOOPING CHANGE TO "BR TST110+2" (674)
5747 010352 012737 000036 000034 11S: MOV #36,2#TRAPVEC ;RESTORE .+2 TO TRAP VECTOR
5748 ;*****
5749 ;TEST 111 IOT
5750 ;*
5751 ;* FORK A SHOULD NOT FAIL.
5752 ;*
5753 ;* IF THE TRAP VECTOR LOGIC FAILS THE TRAP VECTOR COULD COME
5754 ;* OUT TO BE 0, 4, OR 24.
5755 ;*
5756 ;* THE ONLY OTHER POSSIBLE FAILURE WOULD BE STATE TRP.01.
5757 ;* IF THIS STATE FAILS TO LOAD THE DR THE TRAP VECTOR WILL
5758 ;* BE WHATEVER IS IN R4. IF IT FAILS TO LOAD THE BR THE OLD
5759 ;* PS WILL FAIL TO BE STACKED.
5760 ;*
5761 ;* ROM FLOW-14,354,(SVC.00-SVC.90) 355,65,357,360,367,37,25,41,222,300

```



```

5762          005200
5763 010360 005200
5764 010362 012737 010440 000020
5765 010370 012706 001100
5766 010374 012737 010460 000000
5767 010402 012737 010462 000014
5768 010410 012704 000014
5769 010414 012737 010464 000024
5770 010422 012737 010466 000004
5771 010430 012737 000300 177776
5772 010436 000004
5773 010440 042766 177437 000002 15:
5774 010446 022766 000300 000002
5775 010454 001405
5776 010456 000000
5777
5778 010460
5779 010460 000000
5780
5781
5782 010462
5783 010462 000000
5784
5785 010464
5786 010464 000000
5787
5788
5789 010466
5790 010466 000000
5791
5792
5793 010470 016737 170404 177570 SEQENC: MOV $PASS,2$SWR
5794 010476 005737 000042 TST 2$42
5795 010502 001017 BNE 15
5796 010504 005767 170374 TST $ICNT
5797 010510 001407 BEQ 25
5798 010512 022757 000001 000164 CMP #1,$EOPCT
5799 010520 001010 BNE 15
5800 010522 005067 170356 CLR $ICNT
5801 010526 000405 BR 15
5802 010530 005267 170350 25: INC $ICNT
5803 010534 012767 010000 000142 MOV #10000,$EOPCT
5804 010542 022700 000111 15: CMP #111,RO
5805 010546 001443 BEQ 35
5806 010550 012737 011652 000034 MOV $TRAP,2$TRAPVEC
5807 010556 104400 010564 TYPE 65$
5808 010562 000414 BR 64$
5809
5810
5811 010614
5812 010614 010067 170334 MOV RO,$REGD
5813 010620 016746 170330 MOV $REGD,-(SP)
5814 010624 104402 TYPOC
5815 010626 104400 010634 TYPE 67$
5816 010632 000411 BR 66$
5817 010656

```

```

*****
↑TST111: INC RO ; INCREMENT TEST NUMBER
MOV #15,2$20 ; SETUP THE IOT VECTOR
MOV $STACK,SP ; SETUP THE SP
MOV #25,2$0 ; SETUP LOCATION ZERO
MOV #35,2$14 ; SETUP LOCATION 14
MOV #14,R4 ; SETUP R4
MOV #45,2$24 ; SETUP LOCATION 24
MOV #55,2$ERRVEC ; SET UP LOCATION 4
MOV #PR6,2$PSW ; SETUP THE PSW
IOT ; EXECUTE INSTRUCTION UNDER TEST
15: BIC #177437,2$(SP) ; MASK OF THE PRIORITY BITS ON THE OLD PSW
CMP #300,2$(SP) ; DID OLD SP GET STACKED?
BEQ SEQENC ; BRANCH IF YES
HALT ; STATE TRP.01 FAILED TO LOAD BR
; FOR LOOPING CHANGE TO "BR TST111+2" (741)
25: HALT ; EITHER DAPE TV04 DOES NOT GO HIGH
; OR IT DOES NOT GET TO THE ALU.
; FOR LOOPING CHANGE TO "BR TST111+2" (740)
35: HALT ; STATE TRP.01 FILED TO LOAD DR
; FOR LOOPING CHANGE TO "BR TST111+2" (737)
45: HALT ; EITHER IRCD IOT DOES NOT GET TO
; DAPE E7(4) AS A LOW OR E' IS BAD
; FOR LOOPING CHANGE TO "BR TST111+2" (736)
55: HALT ; EITHER IRCD IOT DOES NOT GO LOW
; OR IT DOES NOT GET TO DAPE
; FOR LOOPING CHANGE TO "BR TST111+2" (735)
SEQENC: MOV $PASS,2$SWR ; DISPLAY PASS COUNT IN LIGHTS
TST 2$42 ; ACT 11 OR XXDP LOAD?
BNE 15 ; BRANCH IF YES
TST $ICNT ; CHANGED PASS COUNT YET?
BEQ 25 ; BRANCH IF NO
CMP #1,$EOPCT ; EOP COUNT AT 1 YET?
BNE 15 ; BRANCH IF NO
CLR $ICNT ; CLEAR CHANGED FLAG
BR 15
25: INC $ICNT ; SET CHANGED FLAG
MOV #10000,$EOPCT ; CHANGE PASS COUNT
15: CMP #111,RO ; IS TEST NUMBER OK?
BEQ 35 ; BRANCH IF YES
MOV $TRAP,2$TRAPVEC ; SETUP TRAP VECTOR
TYPE 65$ ; TYPE ASCIZ STRING
BR 64$ ; GET OVER THE ASCIZ
65$: .ASCIZ '<15><12>/TEST NUMBER(RO) IS /'
64$:
MOV RO,$REGD ; SAVE $REGD FOR TYPEOUT
MOV $REGD,-(SP) ; GO TYPE--OCTAL ASCII(ALL DIGITS)
TYPOC ; TYPE ASCIZ STRING
TYPE 67$ ; TYPE ASCIZ STRING
BR 66$ ; GET OVER THE ASCIZ
67$: .ASCIZ '/ SHOULD BE 111<15><12>'
66$:

```

5818 010656 005037 177746
5819
5820
5821
5822
5823
5824
5825
5826
5827
5828
5829
5830
5831
5832
5833 010662
5834 010562 012737 011652 000034
5835 010670 005267 170204
5836 010674 042767 100000 170176
5837 010702 005327
5838 010704 000144
5839 010706 003015
5840 010710 012737
5841 010712 000144
5842 010714 010704
5843 010716 104400 010746
5844 010722 013700 000042
5845 010726 001405
5846 010730 000005
5847 010732 004710
5848 010734 000240
5849 010736 000240
5850 010740 000240
5851 010742
5852 010742 000137 001210
5853 010746 005015 047105 020104
5854 010754 040520 051523
5855 010760 377 377 000
5856 010764
5857
5858
5859
5860
5861
5862
5863
5864
5865
5866
5867
5868
5869
5870
5871
5872
5873

```
35: CLR 2*CONTRL ;ENABLE CACHE
;*****
.MCALL .STYPOCT,.STYPDEC,.STYPE,.STRAP
;*****
.SBTTL END OF PASS ROUTINE

;*INCREMENT THE PASS NUMBER ($PASS)
;*INDICATE END-OF-PROGRAM AFTER 144 PASSES THRU THE PROGRAM
;*TYPE "END PASS"
;*IF THERES A MONITOR GO TO IT
;*IF THERE ISN'T JUMP TO TST1
;*IF IT IS DESIRED TO HAVE A BELL INDICATE THE "END OF PASS" LOCATION
;*SENDMG CAN BE CHANGED TO 7.

SEOP: MOV 2*STRAP,2*TRAPVEC ;SETUP TRAP VECTOR
      INC $PASS ;INCREMENT THE PASS NUMBER
      BIC #100000,$PASS ;DON'T ALLOW A NEG. NUMBER
      DEC (PC)+ ;LOOP?
SECPCT: .WORD 144
      BGT $DOAGN ;;YES
      MOV (PC)+,2(PC)+ ;;RESTORE COUNTER
SENDCT: .WORD 144
      SEOPCT
      TYPE $SENDMG ;;TYPE "END PASS"
SGET42: MOV 2*42,R0 ;;GET MONITOR ADDRESS
      BEQ $DOAGN ;;BRANCH IF NO MONITOR
      RESET ;;CLEAR THE WORLD
SENDAD: JSR PC,(R0) ;;GO TO MONITOR
      NOP ;;SAVE ROOM
      NOP ;;FOR
      NOP ;;ACT11
$DOAGN: JMP 2*TST1 ;;RETURN
SENDMG: .ASCII <15><12>/END PASS/
SENULL: .BYTE -1,-1,0 ;;NULL CHARACTER STRING
      .EVEN
;*****
.SBTTL TYPE ROUTINE

;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
;*NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
;*NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
;*NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
;*
;*CALL:
;*1) USING A TRAP INSTRUCTION
;* TYPE ,MESADR ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
;*OR
;* TYPE
;* MESADR
;*
```

```

5874          ;*2) USING A JSR INSTRUCTION
5875          ;*   MOV   PS,-(SP)          ;; PUSH PROCESSOR STATUS WORD ON THE STACK
5876          ;*   JSR   PC,$TYPE        ;; CALL TYPE ROUTINE
5877          ;*   MESADDR                ;; FIRST ADDRESS OF MESSAGE
5878
5879 010764 105767 170161 $TYPE: TSTB   $TPFLG          ;; IS THERE A TERMINAL?
5880 010770 100002          SPL     1$              ;; BR IF YES
5881 010772 000000          HALT                    ;; HALT HERE IF NO TERMINAL
5882 010774 000407          BR      3$              ;; LEAVE
5883 010776 010046          1$:  MOV   RD,-(SP)          ;; SAVE RD
5884 011000 017600 000002          MOV   22(SP),RD        ;; GET ADDRESS OF ASCIZ STRING
5885 011004 112046          2$:  MOVB  (RD)+,-(SP)        ;; PUSH CHARACTER TO BE TYPED ONTO STACK
5886 011006 001005          BNE   4$              ;; BR IF IT ISN'T THE TERMINATOR
5887 011010 005726          TST   (SP)+          ;; IF TERMINATOR POP IT OFF THE STACK
5888 011012 012600          MOV   (SP)+,RD        ;; RESTORE RD
5889 011014 062716 000002          3$:  ADD   #2,(SP)          ;; ADJUST RETURN PC
5890 011020 000002          RTI                    ;; RETURN
5891 011022 122716 000011          4$:  CMPB  #HT,(SP)          ;; BRANCH IF <HT>
5892 011026 001426          BEQ   8$              ;;
5893 011030 122716 000200          CMPB  #CRLF,(SP)        ;; BRANCH IF NOT
5894 011034 001004          BNE   5$              ;;
5895 011036 005726          TST   (SP)+          ;; POP <CR><LF> EQUIV
5896 011040 104400 001173          TYPE  $CRLF          ;;
5897 011044 000757          BR      2$              ;; GET NEXT CHARACTER
5898 011046 004767 000056          5$:  JSR   PC,$TYPEC        ;; GO TYPE THIS CHARACTER
5899 011052 126726 170072          6$:  CMPB  $FILLC,(SP)+    ;; IS IT TIME FOR FILLER CHARS.?
5900 011056 001352          BNE   2$              ;; IF NO GO GET NEXT CHAR.
5901 011060 016746 170062          MOV   $NULL,-(SP)      ;; GET # OF FILLER CHARS. NEEDED
5902          AND   THE NULL CHAR.
5903 011064 105366 000001          7$:  DECB  1(SP)          ;; DOES A NULL NEED TO BE TYPED?
5904 011070 002770          BLT   6$              ;; BR IF NO--GO POP THE NULL OFF OF STACK
5905 011072 004767 000032          JSR   PC,$TYPEC        ;; GO TYPE A NULL
5906 011076 105367 000072          DECB  $CHARCNT        ;; DON'T COUNT THE NULL AS A CHARACTER
5907 011102 000770          BR      7$              ;; LOOP
5908
5909          ;; HORIZONTAL TAB PROCESSOR
5910
5911 011104 112716 000040          8$:  MOVB  #' (SP)          ;; REPLACE TAB WITH SPACE
5912 011110 004767 000014          9$:  JSR   PC,$TYPEC        ;; TYPE A SPACE
5913 011114 132767 000007 000052          BITB  #7,$CHARCNT      ;; BRANCH IF NOT AT
5914 011122 001372          BNE   9$              ;; TAB STOP
5915 011124 005726          TST   (SP)+          ;; POP SPACE OFF STACK
5916 011126 000726          BR      2$              ;; GET NEXT CHARACTER
5917 011130 105777 170006          $TYPEC: TSTB  2$TPS        ;; WAIT UNTIL PRINTER IS READY
5918 011134 100375          BPL   $TYPEC          ;;
5919 011136 116677 000002 170000          MOVB  2(SP),2$TPB        ;; LOAD CHAR TO BE TYPED INTO DATA REG.
5920 011144 122766 000015 000002          CMPB  #CR,2(SP)        ;; BRANCH IF
5921 011152 001003          BNE   1$              ;; NOT <CR>
5922 011154 105067 000014          CLRB  $CHARCNT        ;;
5923 011160 000406          BR      $TYPEX        ;; EXIT
5924 011162 122766 000012 000002          1$:  CMPB  #LF,2(SP)        ;; BRANCH IF
5925 011170 001402          BEQ   $TYPEX        ;; <LF>
5926 011172 105227          INCB  (PC)+          ;; INC SPACE
5927 011174 000000          $CHARCNT: .WORD 0    ;; COUNT
5928 011176 000207          $TYPEX: RTS   PC
5929

```

```

5930 ;:*****
5931
5932 .SBTTL BINARY TO OCTAL (ASCII) AND TYPE
5933
5934 ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
5935 ;*OCTAL (ASCII) NUMBER AND TYPE IT.
5936 ;*STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
5937 ;*CALL:
5938 ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
5939 ;*      TYPOS    ;;CALL FOR TYPEOUT
5940 ;*      .BYTE   N              ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
5941 ;*      .BYTE   M              ;;M=1 OR 0
5942 ;*                               ;;1=TYPE LEADING ZEROS
5943 ;*                               ;;0=SUPPRESS LEADING ZEROS
5944
5945 ;*STYPON--- ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
5946 ;*STYPOS OR STYPOC
5947 ;*CALL:
5948 ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
5949 ;*      TYPON    ;;CALL FOR TYPEOUT
5950
5951 ;*STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
5952 ;*CALL:
5953 ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
5954 ;*      TYPOC    ;;CALL FOR TYPEOUT
5955
5956 011200 017646 000000          STYPOS: MOV      2(SP),-(SP)      ;; PICKUP THE MODE
5957 011204 116667 000001 000211  MOVVB  1(SP),$OFILL      ;; LOAD ZERO FILL SWITCH
5958 011212 112667 000207          MOVVB  (SP)+,$OMODE+1      ;; NUMBER OF DIGITS TO TYPE
5959 011216 062716 000002          ADD     #2,(SP)          ;; ADJUST RETURN ADDRESS
5960 011222 000406          BR      STYPON
5961 011224 112767 000001 000171  STYPOC: MOVVB  #1,$OFILL      ;; SET THE ZERO FILL SWITCH
5962 011232 112767 000006 000165  MOVVB  #6,$OMODE+1      ;; SET FOR SIX(6) DIGITS
5963 011240 112767 000005 000154  STYPON: MOVVB  #5,$OCNT      ;; SET THE ITERATION COUNT
5964 011246 010346          MOV     R3,-(SP)        ;; SAVE R3
5965 011250 010446          MOV     R4,-(SP)        ;; SAVE R4
5966 011252 010546          MOV     R5,-(SP)        ;; SAVE R5
5967 011254 116704 000145          MOVVB  $OMODE+1,R4      ;; GET THE NUMBER OF DIGITS TO TYPE
5968 011260 005404          NEG     R4
5969 011262 062704 000006          ADD     #6,R4          ;; SUBTRACT IT FOR MAX. ALLOWED
5970 011266 110467 000132          MOVVB  R4,$OMODE      ;; SAVE IT FOR USE
5971 011272 116704 000125          MOVVB  $OFILL,R4      ;; GET THE ZERO FILL SWITCH
5972 011276 016605 000012          MOV     12(SP),R5      ;; PICKUP THE INPUT NUMBER
5973 011302 005003          CLR     R3            ;; CLEAR THE OUTPUT WORD
5974 011304 006105          1$:   ROL     R5          ;; ROTATE MSB INTO "C"
5975 011306 000404          BR      3$           ;; GO DO MSB
5976 011310 006105          2$:   ROL     R5          ;; FORM THIS DIGIT
5977 011312 006105          ROL     R5
5978 011314 006105          ROL     R5
5979 011316 010503          MOV     R5,R3
5980 011320 006103          3$:   ROL     R3          ;; GET LSB OF THIS DIGIT
5981 011322 105367 000076          DECB   $OMODE          ;; TYPE THIS DIGIT?
5982 011326 100016          BPL    7$             ;; BR IF NO
5983 011330 042703 177770          BIC    #177770,R3      ;; GET RID OF JUNK
5984 011334 001002          BNE    4$             ;; TEST FOR 0
5985 011336 005704          TST    R4            ;; SUPPRESS THIS 0?
    
```

```

5986 011340 001403
5987 011342 005204
5988 011344 052703 000060
5989 011350 052703 000040
5990 011354 110367 000040
5991 011360 104400 011420
5992 011364 105367 000032
5993 011370 003347
5994 011372 002402
5995 011374 005204
5996 011376 000744
5997 011400 012605
5998 011402 012604
5999 011404 012603
6000 011406 016666 000002 000004
6001 011414 012616
6002 011416 000002
6003 011420 000
6004 011421 000
6005 011422 000
6006 011423 000
6007 011424 000000
6008
6009
6010
6011
6012
6013
6014
6015
6016
6017
6018
6019
6020
6021 011426
6022 011426 010046
6023 011430 010146
6024 011432 010246
6025 011434 010346
6026 011436 010546
6027 011440 012746 020200
6028 011444 016605 000020
6029 011450 100004
6030 011452 005405
6031 011454 112766 000055 000001
6032 011462 005000
6033 011464 012703 011642
6034 011470 112723 000040
6035 011474 005002
6036 011476 016001 011632
6037 011502 160105
6038 011504 002402
6039 011506 005202
6040 011510 000774
6041 011512 060105

    BEQ 5$
    INC R4
    BIS #'0,R3
    BIS #' ,R3
    MOVB R3,8$
    TYPE 8$
    7$: DECB $OCNT
    BGT 2$
    BLT 6$
    INC R4
    BR 2$
    6$: MOV (SP)+,R5
    MOV (SP)+,R4
    MOV (SP)+,R3
    MOV 2(SP),4(SP)
    MOV (SP)+,(SP)
    RTI
    8$: .BYTE 0
    .BYTE 0
    $OCNT: .BYTE 0
    $DFILL: .BYTE 0
    $OMODE: .WORD 0
    ;;*****
.SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
;*REPLACED WITH SPACES.
;*CALL:
;* MOV NUM,-(SP) ;;PUT THE BINARY NUMBER ON THE STACK
;* TYPDS ;;GO TO THE ROUTINE
$TYPDS:
    MOV R0,-(SP) ;;PUSH R0 ON STACK
    MOV R1,-(SP) ;;PUSH R1 ON STACK
    MOV R2,-(SP) ;;PUSH R2 ON STACK
    MOV R3,-(SP) ;;PUSH R3 ON STACK
    MOV R5,-(SP) ;;PUSH R5 ON STACK
    MOV #20200,-(SP) ;;SET BLANK SWITCH AND SIGN
    MOV 20(SP),R5 ;;GET THE INPUT NUMBER
    BPL 1$ ;;BR IF INPUT IS POS.
    NEG R5 ;;MAKE THE BINARY NUMBER POS.
    MOVB #'-,1(SP) ;;MAKE THE ASCII NUMBER NEG.
    1$: CLR R0 ;;ZERO THE CONSTANTS INDEX
    MOV #SDBLK,R3 ;;SETUP THE OUTPUT POINTER
    MOVB #' ,(R3)+ ;;SET THE FIRST CHARACTER TO A BLANK
    2$: CLR R2 ;;CLEAR THE BCD NUMBER
    MOV $DTBL(R0),R1 ;;GET THE CONSTANT
    3$: SUB R1,R5 ;;FORM THIS BCD DIGIT
    BLT 4$ ;;BR IF DONE
    INC R2 ;;INCREASE THE BCD DIGIT BY 1
    4$: ADD R1,R5 ;;ADD BACK THE CONSTANT

```

```

6042 011514 005702          TST      R2          ;;CHECK IF BCD DIGIT=0
6043 011516 001002          BNE     5$          ;;FALL THROUGH IF 0
6044 011520 105716          TSTB   (SP)        ;;STILL DOING LEADING 0'S?
6045 011522 100407          BMI     7$          ;;BR IF YES
6046 011524 106316          ASLB   (SP)        ;;MSD?
6047 011526 103003          BCC    6$          ;;BR IF NO
6048 011530 116663 000001 177777 MOVB   1(SP),-1(R3) ;;YES--SET THE SIGN
6049 011536 052702 000060 6$:   BIS   #'0,R2      ;;MAKE THE BCD DIGIT ASCII
6050 011542 052702 000040 7$:   BIS   #' ,R2      ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
6051 011546 110223          MOVB   R2,(R3)+    ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
6052 011550 005720          TST   (R0)+        ;;JUST INCREMENTING
6053 011552 020027 000010          CMP   R0,#10      ;;CHECK THE TABLE INDEX
6054 011556 002746          BLT   2$          ;;GO DO THE NEXT DIGIT
6055 011560 003000          BGT   8$          ;;GO TO EXIT
6056 011562 010502          MOV   R5,R2       ;;GET THE LSD
6057 011564 000764          BR    6$          ;;GO CHANGE TO ASCII
6058 011566 105726          TSTB  (SP)+        ;;WAS THE LSD THE FIRST NON-ZERO?
6059 011570 100003          BPL   9$          ;;BR IF NO
6060 011572 116663 177777 177776 MOVB  -1(SP),-2(R3) ;;YES--SET THE SIGN FOR TYPING
6061 011600 105013          CLRB  (R3)        ;;SET THE TERMINATOR
6062 011602 012605          MOV   (SP)+,R5    ;;POP STACK INTO R5
6063 011604 012603          MOV   (SP)+,R3    ;;POP STACK INTO R3
6064 011606 012602          MOV   (SP)+,R2    ;;POP STACK INTO R2
6065 011610 012601          MOV   (SP)+,R1    ;;POP STACK INTO R1
6066 011612 012600          MOV   (SP)+,R0    ;;POP STACK INTO R0
6067 011614 104400 011642          TYPE  $DBLK        ;;NOW TYPE THE NUMBER
6068 011620 016666 000002 000004 MOV   2(SP),4(SP)  ;;ADJUST THE STACK
6069 011626 012616          MOV   (SP)+,(SP)
6070 011630 000002          RTI
6071 011632 023420          SDTBL: 10000.    ;;RETURN TO USER
6072 011634 001750          1000.
6073 011636 000144          100.
6074 011640 000012          10.
6075 011642 000004          $DBLK: .BLKW 4
6076          ;;*****
6077
6078          .SBTTL TRAP DECODER
6079
6080          ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
6081          ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
6082          ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
6083          ;*GO TO THAT ROUTINE.
6084
6085 011652 010046          STRAP: MOV   R0,-(SP) ;;SAVE R0
6086 011654 016600 000002          MOV   2(SP),R0     ;;GET TRAP ADDRESS
6087 011660 005740          TST   -(R0)        ;;BACKUP BY 2
6088 011662 111000          MOVB  (R0),R0      ;;GET RIGHT BYTE OF TRAP
6089 011664 016000 011672          MOV   $TRPAD(R0),R0 ;;INDEX TO TABLE
6090 011670 006200          RTS   R0           ;;GO TO ROUTINE
6091
6092
6093          .SBTTL TRAP TABLE
6094
6095          ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
6096          ;*BY THE "TRAP" INSTRUCTION.
6097

```

6098		
6099		
6100	011672	
6101	011672	010764
6102	011674	011224
6103	011676	011200
6104	011703	011240
6105	011702	011426
6106	011704	011704
6107		000001

```

; ROUTINE
;-----
$TRPAD:
$TYPE      ;; CALL=TYPE      TRAP+0(104400)  TTY TYPEOUT ROUTINE
$TYPOC     ;; CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
$TYPOS     ;; CALL=TYPOS    TRAP+4(104404)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
$TYPON     ;; CALL=TYPON    TRAP+6(104406)  TYPE OCTAL NUMBER (AS PER LAST CALL)
$TYPDS     ;; CALL=TYPDS    TRAP+10(104410) TYPE DECIMAL NUMBER (WITH SIGN)
SUBTAB: .WORD .
        .END

```


IUT66	006722	5053#		
IUT67	006750	5085#		
IUT70	007004	5119#		
IUT71	007062	5165#		
IUT72	007136	5209#		
IUT73	007176	5247#		
IUT74	007250	5288#		
IUT75	007312	5329#		
IUT76	007374	5375#		
IUT77	007436	5409#		
JMPT0	006002	4682	4686#	
JMPIAD	005032	4204	4216#	
JMP2AD	005056	4230	4234#	
KDPAR0=	172360	2325#		
KDPAR1=	172362	2326#		
KDPAR2=	172364	2327#		
KDPAR3=	172366	2328#		
KDPAR4=	172370	2329#		
KDPAR5=	172372	2330#		
KDPAR6=	172374	2331#		
KDPAR7=	172376	2332#		
KDPDR0=	172320	2303#		
KDPDR1=	172322	2304#		
KDPDR2=	172324	2305#		
KDPDR3=	172326	2306#		
KDPDR4=	172330	2307#		
KDPDR5=	172332	2308#		
KDPDR6=	172334	2309#		
KDPDR7=	172336	2310#		
KERSTK=	001100	2042#		
KIPAR0=	172340	2314#		
KIPAR1=	172342	2315#		
KIPAR2=	172344	2316#		
KIPAR3=	172346	2317#		
KIPAR4=	172350	2318#		
KIPAR5=	172352	2319#		
KIPAR6=	172354	2320#		
KIPAR7=	172356	2321#		
KIPDR0=	172300	2292#		
KIPDR1=	172302	2293#		
KIPDR2=	172304	2294#		
KIPDR3=	172306	2295#		
KIPDR4=	172310	2296#		
KIPDR5=	172312	2297#		
KIPDR6=	172314	2298#		
KIPDR7=	172316	2299#		
KSP	=%000006	2076#		
LF	= 000012	2056#	5924	5930
LOADRS=	177740	2166#		
MAINT =	177750	2170#		
MAPHO =	170202	2409#		
MAPHO0=	170202	2345#	2409	
MAPHO1=	170206	2347#	2411	
MAPHO2=	170212	2349#	2413	
MAPHO3=	170216	2351#	2415	
MAPHO4=	170222	2353#	2417	

MAPH05=	170226	2355#	2419
MAPH06=	170232	2357#	2421
MAPH07=	170236	2359#	2423
MAPH1 =	170206	2411#	
MAPH10=	170242	2361#	
MAPH11=	170246	2363#	
MAPH12=	170252	2365#	
MAPH13=	170256	2367#	
MAPH14=	170262	2369#	
MAPH15=	170266	2371#	
MAPH16=	170272	2373#	
MAPH17=	170276	2375#	
MAPH2 =	170212	2413#	
MAPH20=	170302	2377#	
MAPH21=	170306	2379#	
MAPH22=	170312	2381#	
MAPH23=	170316	2383#	
MAPH24=	170320	2385#	
MAPH25=	170326	2387#	
MAPH26=	170332	2389#	
MAPH27=	170336	2391#	
MAPH3 =	170216	2415#	
MAPH30=	170342	2393#	
MAPH31=	170346	2395#	
MAPH32=	170352	2397#	
MAPH33=	170356	2399#	
MAPH34=	170362	2401#	
MAPH35=	170366	2403#	
MAPH36=	170372	2405#	
MAPH37=	170376	2407#	
MAPH4 =	170222	2417#	
MAPH5 =	170226	2419#	
MAPH6 =	170232	2421#	
MAPH7 =	170236	2423#	
MAPL0 =	170200	2408#	
MAPL00=	170200	2344#	2408
MAPL01=	170204	2346#	2410
MAPL02=	170210	2348#	2412
MAPL03=	170214	2350#	2414
MAPL04=	170220	2352#	2416
MAPL05=	170224	2354#	2418
MAPL06=	170230	2356#	2420
MAPL07=	170234	2358#	2422
MAPL1 =	170204	2410#	
MAPL10=	170240	2360#	
MAPL11=	170244	2362#	
MAPL12=	170250	2364#	
MAPL13=	170254	2366#	
MAPL14=	170260	2368#	
MAPL15=	170264	2370#	
MAPL16=	170270	2372#	
MAPL17=	170274	2374#	
MAPL2 =	170210	2412#	
MAPL20=	170300	2376#	
MAPL21=	170304	2378#	
MAPL22=	170310	2380#	

SW03	=	000010	2104#	2114				
SW04	=	000020	2103#	2113				
SW05	=	000040	2102#	2112				
SW06	=	000100	2101#	2111				
SW07	=	000200	2100#	2110				
SW08	=	000400	2099#	2109				
SW09	=	001000	2098#	2108				
SW1	=	000002	2116#					
SW10	=	002000	2097#					
SW11	=	004000	2096#					
SW12	=	010000	2095#					
SW13	=	020000	2094#					
SW14	=	040000	2093#					
SW15	=	100000	2092#					
SW2	=	000004	2115#					
SW3	=	000010	2114#					
SW4	=	000020	2113#					
SW5	=	000040	2112#					
SW6	=	000100	2111#					
SW7	=	000200	2110#					
SW8	=	000400	2109#					
SW9	=	001000	2108#					
SXTRO		002024	3022	3023	3024	3026#	3035	
SXT2		002040	3034#	3048				
SYNC24		003332	3540#					
SYNC25		003426	3590#					
SYNC26		003522	3657#					
SYNC27		003652	3725#					
SYNC31		004302	3912#					
SYNC32		004424	3970#					
SYNC33		004502	4014#					
SYNC34		004626	4083#					
SYNC35		004706	4119#					
SYNC36		004730	4137#					
SYNC37		004752	4160#					
SYNC40		005016	4206#					
SYNC41		005052	4231#					
SYNC42		005074	4250#					
SYNC43		005124	4285#	4303#				
SYNC44		005216	4344#					
SYNC45		005300	4384#					
SYNC46		005326	4409#					
SYNC47		005412	4459#					
SYNC50		005456	4493#					
SYNC51		005534	4540#					
SYNC52		005630	4596#					
SYNC53		005704	4633#					
SYNC55		005774	4683#					
SYNC56		006022	4711#					
SYNC57		006056	4736#					
SYNC60		006114	4765#					
SYNC61		006142	4804#					
SYNC62		006342	4857	4870#				
SYNC63		006462	4928#					
SYNC64		006510	4949#					
SYNC65		006570	5001#					

COMA	3673#														
COMAA	4388#														
COMB	3676#	3677													
COMBB	4500#														
COMC	3697#	3698													
COMCC	5125#														
COMD	3940#														
COMDD	5785#	5786													
COME	3927#	3928	4057#	4058											
COMEE	5778#	5779													
COMENT	4570#	4571													
COMFF	5789#	5790													
COMG	4292#	4293													
COMGG	5708#	5709													
COMH	4292#	4297													
COMHH	5713#	5714													
COMI	4351#														
COMII	5717#	5718													
COMJ	4504#	4505													
COMJJ	5721#	5722													
COMK	4563#														
COMKK	5725#	5726													
COML	4686#	4687													
COMLL	5728#	5729													
COMMEN	1#	2427#													
COMMM	5731#	5732													
COMN	5171#														
COMNN	5740#	5741													
COMO	5253#														
COMP	5415#	5697#													
COMQ	5704#														
COMX	5617#														
COMY	5610#														
COMZ	5655#														
DOC	2011#	2538	2539	2540	2542	2543	2545	2547	2549	2551	2553	2556	2557	2559	2561
	2562	2563	2564	2566	2567	2569	2570	2572	2574	2575	2576	2578	2580	2582	2584
	2588	2589	2591	2593	2595	2597	2599	2601	2604	2605	2606	2608	2610	2612	2614
	2616	2617	2619	2620											
DOCEXP	2011#	2538	2539	2540	2542	2543	2545	2547	2549	2551	2553	2556	2557	2559	2561
	2562	2563	2564	2566	2567	2569	2570	2572	2574	2575	2576	2578	2580	2582	2584
	2588	2589	2591	2593	2595	2597	2599	2601	2604	2605	2606	2608	2610	2612	2614
	2616	2617	2619	2620											
ENDCOM	1#	2427#													
ERROR	2045#														
ERRORD	2011#	2646	2666	2672	2692	2698	2717	2723	2742	2748	2770	2787	2793	2811	2817
	2834	2840	2856	2876	2888	2900	2911	2925	2937	2948	2960	2972	2987	2999	3013
	3026	3034	3047	3067	3070	3082	3095	3107	3122	3135	3147	3178	3180	3186	3197
	3199	3205	3216	3219	3224	3232	3236	3240	3244	3251	3255	3267	3271	3277	3281
	3293	3297	3303	3307	3319	3323	3329	3333	3345	3349	3355	3359	3371	3375	3381
	3385	3397	3401	3407	3411	3446	3450	3455	3459	3464	3468	3473	3477	3482	3486
	3491	3495	3559	3561	3567	3595	3610	3612	3617	3619	3624	3673	3676	3684	3688
	3694	3697	3701	3707	3738	3740	3756	3758	3784	3790	3798	3803	3812	3819	3828
	3835	3844	3851	3859	3862	3871	3880	3889	3925	3927	3931	3940	3948	3974	3983
	4024	4026	4036	4041	4043	4055	4057	4061	4093	4097	4099	4123	4142	4167	4170
	4211	4213	4219	4236	4290	4292	4296	4308	4318	4351	4354	4360	4365	4388	4416
	4418	4465	4467	4500	4504	4553	4557	4563	4567	4570	4603	4605	4610	4639	4643

	3283	3309	3335	3361	3387	3413	3520	3570	3627	3709	3761	3891	3952	3985	4046
	4102	4125	4144	4181	4221	4238	4255	4322	4367	4391	4429	4470	4508	4576	4613
	4648	4670	4692	4717	4750	4790	4816	4915	4934	4961	5037	5070	5102	5150	5189
	5229	5270	5307	5358	5394	5432	5471	5507	5536	5554	5573	5589	5624	5666	5748
POP	18	2427	6062												
PUSH	18	2427	6021												
SAVERD	2011														
SCOPE	2046														
SETTRA	6092	6102	6103	6104	6105										
SETUP	18	2427													
SKIP	18	2427	2645	2671	2697	2722	2747	2769	2792	2816	2839	2855	3223	3254	3280
	3306	3332	3358	3384	3410	3565	3623	3706	3748	3888	3982	4035	4086	4122	4141
	4165	4217	4235	4317	4364	4387	4412	4464	4496	4544	4609	4636	4667	4714	4747
	4769	4809	4904	4931	4957	5034	5056	5088	5138	5168	5211	5250	5291	5346	5378
	5412	5455	5491	5524	5550	5570	5585	5605	5654	5703	5775				
SLASH	18	2427													
SPACE	2427														
STARS	18	2427	2440	2466	2514	2531	2622	2629	2638	2651	2660	2676	2685	2701	2710
	2726	2735	2751	2756	2764	2773	2781	2797	2805	2820	2828	2843	2850	2859	2868
	3152	3169	3257	3262	3283	3288	3309	3314	3335	3340	3361	3366	3387	3392	3413
	3419	3509	3520	3529	3547	3551	3570	3584	3627	3654	3709	3722	3761	3777	3792
	3805	3821	3837	3853	3864	3873	3882	3891	3908	3952	3966	3985	4004	4009	4013
	4046	4078	4102	4113	4125	4132	4144	4155	4174	4178	4181	4201	4221	4228	4238
	4247	4255	4278	4322	4338	4367	4378	4391	4403	4423	4426	4429	4446	4470	4486
	4508	4528	4576	4589	4613	4625	4648	4657	4670	4680	4692	4705	4717	4728	4750
	4760	4774	4778	4783	4787	4790	4799	4816	4836	4909	4912	4915	4921	4934	4943
	4961	4990	5037	5047	5062	5066	5070	5079	5095	5098	5102	5113	5144	5147	5150
	5160	5179	5186	5189	5199	5221	5226	5229	5240	5261	5266	5270	5280	5301	5304
	5307	5322	5351	5355	5358	5368	5387	5391	5394	5403	5423	5429	5432	5442	5463
	5467	5471	5481	5501	5504	5507	5514	5529	5533	5536	5542	5554	5560	5573	5577
	5589	5599	5624	5638	5661	5665	5666	5674	5748	5762	5819	5821	5857	5930	6008
TRMTRP	6092														
TYPBIN	18	2427													
TYPDEC	18	2427													
TYPNAM	18	2427													
TYPNUM	18	2427													
TYPOCS	18	2427													
TYPOCT	18	2427	5812												
TYPTXT	18	2427	5807	5814											
SIUT	2011	3542	3592	3659	3726	3971	4015	4084	4120	4139	4161	4207	4232	4251	4286
	4304	4345	4385	4410	4460	4494	4541	4597	4634	4684	4712	4737	4766	4805	4872
	4929	4950	5002	5053	5085	5119	5165	5209	5247	5288	5329	5375	5409	5452	5488
	5522	5548	5568	5645											
SSAVEA	2011														
SSYNC	2011	3540	3590	3657	3725	3912	3970	4014	4083	4119	4137	4160	4206	4231	4250
	4285	4303	4344	4384	4409	4459	4493	4540	4596	4633	4683	4711	4736	4765	4804
	4870	4928	4949	5001	5052	5084	5118	5164	5208	5246	5287	5327	5374	5408	5451
	5487	5521	5547	5567	5644										
SSCMRE	2466	2501	2502	2503											
SSCMTH	2466	2504	2505	2506	2507										
SSESCA	18	2427													
SSNEWT	18	2427	2629	2651	2676	2701	2726	2756	2773	2797	2820	2843	2859	3152	3257
	3283	3309	3335	3361	3387	3413	3520	3570	3627	3709	3761	3891	3952	3985	4046
	4102	4125	4144	4181	4221	4238	4255	4322	4367	4391	4429	4470	4508	4576	4613
	4648	4670	4692	4717	4750	4790	4816	4915	4934	4961	5037	5070	5102	5150	5189

CLRB	5206	5922	6061															
CLV	3116																	
CLZ	2870	2882	2954	3041	3076	3089	3129	3141	3658	4138								
CMP	3142	3265	3269	3275	3279	3291	3295	3301	3305	3317	3321	3327	3331	3343	3347			
	3353	3357	3369	3373	3379	3383	3395	3399	3405	3409	3444	3448	3453	3457	3462			
	3466	3471	3475	3480	3484	3489	3493	3604	3608	3615	3666	3692	3686	3736	3747			
	3754	3920	3923	3936	4017	4022	4030	4039	4053	4095	4140	4164	4209	4234	4347			
	4413	4637	4641	4746	4886	4890	4952	5014	5018	5033	5055	5121	5123	5129	5167			
	5169	5212	5251	5292	5337	5341	5345	5377	5379	5411	5413	5492	5569	5604	5608			
	5615	5653	5691	5695	5702	5774	5798	5804	6053									
CMPB	4598	4601	4608	4930	5210	5523	5549	5584	5891	5893	5899	5920	5924					
COM	2895	3273	3274	3299	3300	3325	3326	3351	3352	3377	3378	3782	3788	3801	3810			
	3817	3826	3833	3842	3849	3869	3878	3887	5082									
DEC	2932	3018	3039	3603	3753	4448	4449	4452	4453	4535	4536	5207	5837					
DECB	5903	5906	5981	5992														
EMT	2045	5688																
HALT	2435	2647	2667	2673	2693	2699	2718	2724	2743	2749	2771	2788	2794	2812	2818			
	2835	2841	2857	2877	2889	2901	2912	2926	2938	2949	2961	2973	2988	3000	3014			
	3027	3035	3048	3068	3071	3083	3096	3108	3123	3136	3148	3178	3181	3186	3197			
	3200	3205	3216	3219	3224	3232	3236	3240	3244	3251	3255	3267	3271	3277	3281			
	3293	3297	3303	3307	3319	3323	3329	3333	3345	3349	3355	3359	3371	3375	3381			
	3385	3397	3401	3407	3411	3446	3450	3455	3459	3464	3468	3473	3477	3482	3486			
	3491	3495	3559	3562	3567	3595	3610	3613	3617	3620	3624	3673	3677	3684	3688			
	3695	3698	3702	3707	3738	3741	3756	3759	3784	3790	3798	3803	3812	3819	3828			
	3835	3844	3851	3859	3862	3871	3880	3889	3925	3928	3932	3940	3948	3974	3983			
	4024	4027	4036	4041	4044	4055	4058	4062	4093	4097	4100	4123	4142	4167	4170			
	4211	4214	4219	4236	4290	4293	4297	4308	4319	4351	4355	4360	4365	4388	4416			
	4419	4465	4468	4500	4505	4553	4557	4563	4568	4571	4603	4606	4610	4639	4643			
	4646	4664	4668	4687	4715	4740	4744	4748	4770	4811	4814	4879	4883	4888	4892			
	4698	4905	4932	4954	4958	5008	5016	5020	5024	5035	5057	5089	5125	5131	5134			
	5139	5171	5175	5214	5217	5253	5257	5294	5297	5332	5335	5339	5343	5347	5381			
	5384	5415	5419	5456	5459	5494	5497	5525	5551	5571	5586	5610	5617	5622	5649			
	5655	5689	5693	5697	5704	5709	5714	5718	5722	5726	5729	5732	5741	5745	5776			
	5779	5783	5786	5790	5881													
INC	2943	2979	3032	3087	3100	3127	3171	3175	3183	3190	3194	3202	3209	3213	3221			
	3229	3248	3421	3424	3427	3430	3433	3436	3439	3442	3498	3502	3511	3515	3517			
	3530	3532	3535	3538	3552	3553	3585	3588	3601	3655	3723	3734	3735	3746	3778			
	3909	3967	4005	4032	4079	4091	4114	4133	4156	4202	4229	4248	4279	4302	4312			
	4339	4342	4379	4383	4404	4447	4487	4491	4529	4590	4595	4626	4658	4681	4706			
	4710	4729	4761	4768	4800	4837	4922	4927	4944	4991	4996	5048	5080	5114	5116			
	5161	5200	5241	5281	5323	5369	5404	5444	5482	5515	5543	5561	5578	5600	5639			
	5675	5763	5802	5835	5987	5995	6039											
INCB	5926																	
IOT	2046	5772																
JMP	2439	4208	4233	4252	4685	5852												
JSR	5330	5847	5898	5905	5912													
MOV	2628	3176	3184	3195	3203	3214	3222	3234	3242	3253	3443	3452	3461	3470	3479			
	3488	3518	3545	3548	3549	3555	3660	3665	3670	3727	3731	3732	3733	3743	3744			
	3750	3751	3780	3786	3794	3795	3807	3814	3823	3830	3839	3846	3855	3866	3875			
	3884	3910	3914	3918	3919	3922	3934	3935	3938	3939	3943	3945	3946	3950	3951			
	3968	3972	3976	3977	3978	3980	4006	4007	4008	4011	4012	4016	4019	4020	4021			
	4029	4034	4038	4052	4064	4080	4081	4082	4089	4090	4116	4117	4118	4135	4136			
	4158	4159	4166	4169	4204	4205	4230	4249	4280	4281	4340	4343	4357	4362	4380			
	4381	4382	4405	4406	4407	4415	4450	4451	4454	4455	4456	4458	4461	4462	4488			
	4489	4490	4492	4497	4530	4531	4532	4533	4534	4538	4539	4545	4556	4559	4592			
	4594	4600	4627	4628	4629	4631	4659	4660	4661	4662	4682	4707	4708	4709	4730			

	4731	4732	4734	4735	4738	4762	4801	4802	4803	4838	4839	4840	4841	4842	4843
	4844	4845	4846	4847	4848	4852	4853	4854	4855	4858	4859	4860	4866	4868	4873
	4874	4885	4923	4925	4926	4945	4946	4948	4951	4992	4993	4994	4995	4998	4999
	5004	5005	5010	5013	5026	5027	5049	5083	5117	5162	5163	5166	5201	5202	5203
	5204	5242	5243	5244	5248	5282	5283	5284	5285	5289	5324	5325	5326	5370	5371
	5372	5376	5406	5407	5410	5445	5446	5447	5449	5450	5453	5493	5484	5485	5489
	5516	5517	5518	5519	5520	5544	5545	5546	5562	5563	5564	5565	5566	5579	5580
	5581	5582	5583	5601	5602	5606	5640	5641	5642	5643	5651	5676	5677	5678	5679
	5680	5681	5682	5683	5684	5685	5686	5736	5737	5738	5747	5764	5765	5766	5767
	5768	5769	5770	5771	5793	5803	5806	5811	5812	5834	5840	5844	5883	5884	5888
	5901	5956	5964	5965	5966	5972	5979	5997	5998	5999	6000	6001	6022	6023	6024
	6025	6026	6027	6028	6033	6036	6056	6062	6063	6064	6065	6066	6068	6069	6085
	6086	6089													
MOV8	3057	3745	3752	4924	5050	5051	5205	5885	5911	5919	5957	5958	5961	5962	5963
	5967	5970	5971	5990	6031	6034	6048	6051	6060	6088					
NEG	3593	5120	5137	5968	6030										
NOP	3554	3725	3912	3970	4014	4083	4119	4160	4206	4231	4250	4253	4254	4285	4303
	4344	4384	4409	4459	4493	4540	4596	4633	4683	4711	4736	4765	4804	4928	4949
	5052	5084	5118	5164	5208	5246	5287	5374	5408	5451	5487	5521	5547	5567	5644
	5647	5848	5849	5850											
RESET	5846														
ROL	2967	3422	3423	3425	3426	3428	3429	3431	3432	3434	3435	3437	3438	3440	3441
	3499	3500	3501	3503	3504	3505	3506	3507	3508	3512	3513	3514	3516	3533	3534
	3536	3537	3539	3705	5974	5976	5977	5978	5980						
ROLB	4713														
ROR	2920	2966	2978	3005	3053	3113	3114	3557	5028						
RORB	5012	5054													
RTI	5646	5890	6002	6070											
RTS	4663	5928	6090												
RTT	3550														
SBC	2994														
SCC	2869	2881	2905	2930	2942	2953	2980	2992	3006	3040	3055	3075	3088	3101	3115
	3128	3140	5328	5603	5607	5614	5687								
SEC	2661	2916	2965	2977	3004	3052	3112	3856	4547	4871					
SEN	2738	2831	2919	3030	4764										
SEV	2688	2784	2851	2894	3020	3591									
SEZ	2713	2808	2894	3063	3541	3591	3979	4033	5000	5136					
SQB	4287	4305	4313												
SUB	3130	6037													
SWAB	3008	3054	3681	4593	5011										
SXT	3021	3031	4767												
TRAP	5739	6092	6102	6103	6104	6105									
TST	2883	3064	3172	3191	3210	3230	3238	3249	3606	3622	3671	4085	4162	4216	4288
	4306	4314	4316	4349	4358	4411	4463	4666	4742	4807	4809	4856	4877	4881	4896
	4956	5006	5087	5249	5290	5454	5490	5794	5796	5887	5895	5915	5985	6042	6052
	6087														
TSTB	4386	4862	4900	5879	5917	6044	6058								
XOR	3042	5086													
.ASCII	2508	2509	5853												
.ASCIZ	2510	5810	5817												
.BLKW	6075														
.BYTE	2477	2478	2483	2484	2495	2496	2497	2498	5855	6003	6004	6005	6006		
.DSABL	2011														
.ENABL	1	2011													
.ENC	6107														
.ENDC	2031	2037	2426	2428	2440	2441	2463	2465	2467	2474	2499	2504	2508	2514	2515

2532	2623	2630	2631	2638	2639	2640	2646	2650	2651	2652	2653	2660	2661	2662
2668	2669	2672	2675	2676	2677	2678	2685	2686	2687	2694	2695	2698	2700	2701
2702	2703	2710	2711	2712	2719	2720	2723	2725	2726	2727	2728	2735	2736	2737
2744	2745	2748	2750	2751	2752	2757	2758	2764	2765	2766	2770	2772	2773	2774
2775	2781	2782	2783	2789	2790	2793	2796	2797	2798	2799	2805	2806	2807	2813
2814	2817	2819	2820	2821	2822	2828	2829	2830	2836	2837	2840	2842	2843	2844
2845	2850	2851	2852	2856	2858	2859	2860	2861	2868	2869	2870	2878	2879	2890
2891	2902	2903	2913	2914	2927	2928	2939	2940	2950	2951	2962	2963	2974	2975
2989	2990	3001	3002	3015	3016	3028	3029	3036	3037	3049	3050	3069	3070	3072
3073	3084	3085	3097	3098	3109	3110	3124	3125	3137	3138	3149	3150	3153	3154
3169	3170	3171	3179	3180	3182	3183	3187	3188	3198	3199	3201	3202	3206	3207
3217	3218	3220	3221	3224	3225	3226	3233	3234	3237	3238	3241	3242	3245	3246
3252	3253	3255	3256	3257	3258	3259	3262	3263	3264	3268	3269	3272	3273	3278
3279	3281	3282	3283	3284	3285	3288	3289	3290	3294	3295	3298	3299	3304	3305
3307	3308	3309	3310	3311	3314	3315	3316	3320	3321	3324	3325	3330	3331	3333
3334	3335	3336	3337	3340	3341	3342	3346	3347	3350	3351	3356	3357	3359	3360
3361	3362	3363	3366	3367	3368	3372	3373	3376	3377	3382	3383	3385	3386	3387
3388	3389	3392	3393	3394	3398	3399	3402	3403	3408	3409	3411	3412	3413	3414
3415	3419	3420	3421	3447	3448	3451	3452	3456	3457	3460	3461	3465	3466	3469
3470	3474	3475	3478	3479	3483	3484	3487	3488	3492	3493	3496	3497	3510	3521
3522	3529	3530	3531	3541	3548	3552	3560	3561	3563	3564	3567	3568	3569	3571
3572	3584	3585	3586	3591	3596	3597	3611	3612	3614	3615	3618	3619	3621	3622
3624	3625	3626	3628	3629	3654	3655	3656	3658	3675	3676	3680	3681	3685	3686
3689	3690	3696	3697	3700	3701	3703	3704	3707	3708	3709	3710	3711	3722	3723
3724	3725	3726	3739	3740	3742	3743	3749	3757	3758	3760	3761	3762	3763	3777
3778	3779	3785	3786	3791	3792	3793	3799	3800	3804	3805	3806	3813	3814	3820
3821	3822	3829	3830	3836	3837	3838	3845	3846	3852	3853	3854	3860	3861	3863
3864	3865	3872	3873	3874	3881	3882	3883	3889	3890	3891	3892	3893	3908	3909
3910	3912	3913	3926	3927	3930	3931	3933	3934	3942	3943	3949	3950	3953	3954
3966	3967	3968	3970	3971	3975	3976	3983	3984	3985	3986	3987	4004	4005	4006
4010	4014	4015	4025	4026	4028	4029	4036	4037	4038	4042	4043	4045	4046	4047
4048	4056	4057	4060	4061	4063	4064	4078	4079	4080	4083	4084	4088	4094	4095
4098	4099	4101	4102	4103	4104	4113	4114	4115	4119	4120	4123	4124	4125	4126
4127	4132	4133	4134	4138	4142	4143	4144	4145	4146	4155	4156	4157	4160	4161
4166	4168	4169	4171	4172	4175	4179	4182	4183	4201	4202	4203	4206	4207	4212
4213	4215	4216	4218	4220	4221	4222	4223	4228	4229	4230	4231	4232	4236	4237
4238	4239	4240	4247	4248	4249	4250	4251	4256	4257	4278	4279	4280	4285	4286
4291	4292	4295	4296	4300	4301	4303	4304	4309	4310	4318	4320	4321	4323	4324
4338	4339	4340	4344	4345	4353	4354	4356	4357	4361	4362	4365	4366	4367	4368
4369	4378	4379	4380	4384	4385	4388	4390	4391	4392	4393	4403	4404	4405	4409
4410	4413	4417	4418	4420	4421	4424	4427	4430	4431	4446	4447	4448	4459	4460
4465	4466	4467	4469	4470	4471	4472	4486	4487	4488	4493	4494	4497	4503	4504
4507	4508	4509	4510	4528	4529	4530	4540	4541	4545	4554	4555	4558	4559	4566
4567	4569	4570	4575	4576	4577	4578	4589	4590	4591	4596	4597	4604	4605	4607
4608	4610	4611	4612	4614	4615	4625	4626	4627	4633	4634	4637	4640	4641	4644
4645	4647	4648	4649	4650	4657	4658	4659	4665	4666	4668	4669	4670	4671	4672
4680	4681	4682	4683	4684	4690	4691	4693	4694	4705	4706	4707	4711	4712	4715
4716	4717	4718	4719	4728	4729	4730	4736	4737	4741	4742	4745	4746	4748	4749
4750	4751	4752	4760	4761	4762	4765	4766	4770	4771	4772	4775	4779	4784	4788
4791	4792	4799	4800	4801	4804	4805	4809	4812	4813	4815	4816	4817	4818	4836
4837	4838	4871	4880	4881	4884	4885	4889	4890	4894	4895	4899	4900	4905	4906
4907	4910	4913	4916	4917	4921	4922	4923	4928	4929	4932	4933	4934	4935	4936
4943	4944	4945	4949	4950	4955	4956	4958	4959	4960	4962	4963	4990	4991	4992
5002	5009	5010	5017	5018	5021	5022	5025	5026	5035	5036	5037	5038	5039	5047
5048	5049	5052	5053	5057	5058	5059	5063	5067	5071	5072	5079	5080	5081	5084
5085	5089	5090	5091	5096	5099	5103	5104	5113	5114	5115	5118	5119	5128	5129

	5132	5133	5135	5136	5139	5140	5141	5145	5148	5151	5152	5160	5161	5162	5164
	5165	5169	5173	5174	5176	5177	5180	5197	5190	5191	5199	5200	5201	5208	5209
	5212	5215	5216	5218	5219	5222	5227	5230	5231	5240	5241	5242	5246	5247	5251
	5255	5256	5258	5259	5262	5267	5271	5272	5280	5281	5282	5287	5288	5292	5295
	5296	5298	5299	5302	5305	5308	5309	5322	5323	5324	5328	5333	5334	5336	5337
	5340	5341	5344	5345	5347	5348	5349	5352	5356	5359	5360	5368	5369	5370	5374
	5375	5379	5382	5383	5385	5386	5388	5392	5395	5396	5403	5404	5405	5408	5409
	5413	5417	5418	5420	5421	5424	5430	5433	5434	5443	5444	5445	5451	5452	5456
	5457	5458	5460	5461	5464	5468	5472	5473	5481	5482	5483	5487	5488	5492	5495
	5496	5498	5499	5502	5505	5508	5509	5514	5515	5516	5521	5522	5525	5526	5527
	5530	5534	5537	5538	5542	5543	5544	5547	5548	5551	5552	5553	5555	5556	5560
	5561	5562	5567	5568	5571	5572	5573	5574	5575	5577	5578	5579	5586	5587	5588
	5590	5591	5599	5600	5601	5606	5612	5613	5620	5621	5623	5624	5625	5626	5638
	5639	5640	5644	5645	5650	5651	5655	5658	5659	5662	5666	5667	5668	5674	5675
	5676	5690	5691	5694	5695	5700	5701	5704	5707	5708	5712	5713	5716	5717	5720
	5721	5724	5725	5727	5728	5730	5731	5733	5734	5743	5744	5746	5747	5749	5750
	5762	5763	5764	5776	5777	5778	5781	5782	5784	5785	5788	5789	5792	5793	5810
	5817	5820	5822	5825	5827	5828	5830	5832	5835	5839	5842	5843	5844	5846	5852
	5853	5855	5856	5858	5931	6009	6077	6086	6089	6091	6101	6102	6103	6104	6105
	6106														
.EQUIV	2045	2046	2043	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2108
	2109	2110	2111	2112	2113	2114	2115	2116	2117	2136	2137	2138	2139	2140	2141
	2142	2143	2144	2145	2197	2198	2199	2200	2408	2409	2410	2411	2412	2413	2414
	2415	2416	2417	2418	2419	2420	2421	2422	2423						
.EVEN	5810	5817	5856												
.IF	2027	2037	2426	2428	2435	2440	2462	2464	2466	2473	2499	2504	2508	2511	2514
	2531	2622	2629	2631	2638	2640	2645	2647	2650	2651	2653	2660	2662	2667	2668
	2671	2673	2675	2676	2678	2685	2687	2693	2694	2697	2699	2700	2701	2703	2710
	2712	2718	2719	2722	2724	2725	2726	2728	2735	2737	2743	2744	2747	2749	2750
	2751	2756	2758	2764	2766	2769	2771	2772	2773	2775	2781	2783	2788	2789	2792
	2794	2796	2797	2799	2805	2807	2812	2813	2816	2818	2819	2820	2822	2828	2830
	2835	2836	2839	2841	2842	2843	2845	2850	2852	2855	2857	2858	2859	2861	2868
	2870	2877	2878	2889	2890	2901	2902	2912	2913	2926	2927	2938	2939	2949	2950
	2961	2962	2973	2974	2988	2989	3000	3001	3014	3015	3027	3028	3035	3035	3048
	3049	3068	3069	3071	3072	3083	3084	3096	3097	3108	3109	3123	3124	3136	3137
	3148	3149	3152	3154	3169	3171	3178	3179	3181	3182	3186	3187	3197	3198	3200
	3201	3205	3206	3216	3217	3219	3220	3223	3224	3225	3232	3233	3236	3237	3240
	3241	3244	3245	3251	3252	3254	3255	3256	3257	3259	3262	3264	3267	3268	3271
	3272	3277	3278	3280	3281	3282	3283	3285	3288	3290	3293	3294	3297	3298	3303
	3304	3306	3307	3308	3309	3311	3314	3316	3319	3320	3323	3324	3329	3330	3332
	3333	3334	3335	3337	3340	3342	3345	3346	3349	3350	3355	3356	3358	3359	3360
	3361	3363	3366	3368	3371	3372	3375	3376	3381	3382	3384	3385	3386	3387	3389
	3392	3394	3397	3398	3401	3402	3407	3408	3410	3411	3412	3413	3415	3419	3421
	3446	3447	3450	3451	3455	3456	3459	3460	3464	3465	3468	3469	3473	3474	3477
	3478	3482	3483	3486	3487	3491	3492	3495	3496	3509	3520	3522	3529	3531	3540
	3541	3547	3551	3559	3560	3562	3563	3566	3567	3568	3570	3572	3584	3586	3590
	3591	3595	3596	3610	3611	3613	3614	3617	3618	3620	3621	3623	3624	3625	3627
	3629	3654	3656	3657	3658	3673	3675	3677	3680	3684	3685	3688	3689	3695	3696
	3698	3700	3702	3703	3706	3707	3708	3709	3711	3722	3724	3725	3738	3739	3741
	3742	3748	3756	3757	3759	3760	3761	3763	3777	3779	3784	3785	3790	3791	3792
	3798	3799	3803	3804	3805	3812	3813	3819	3820	3821	3828	3829	3835	3836	3837
	3844	3845	3851	3852	3853	3859	3860	3862	3863	3864	3871	3872	3873	3880	3881
	3882	3888	3889	3890	3891	3893	3908	3910	3912	3925	3926	3928	3930	3932	3933
	3940	3942	3948	3949	3952	3954	3966	3968	3970	3974	3975	3982	3983	3984	3985
	3987	4004	4006	4009	4013	4014	4024	4025	4027	4028	4035	4036	4037	4041	4042
	4044	4045	4046	4048	4055	4056	4058	4060	4062	4063	4078	4080	4083	4087	4093

4094	4097	4098	4100	4101	4102	4104	4113	4115	4119	4122	4123	4124	4125	4127
4132	4134	4137	4138	4141	4142	4143	4144	4146	4155	4157	4160	4165	4167	4168
4170	4171	4174	4178	4181	4183	4201	4203	4206	4211	4212	4214	4215	4217	4219
4220	4221	4223	4228	4230	4231	4235	4236	4237	4238	4240	4247	4249	4250	4255
4257	4278	4280	4285	4290	4291	4293	4295	4297	4300	4303	4308	4309	4317	4319
4320	4322	4324	4338	4340	4344	4351	4353	4355	4356	4360	4361	4364	4365	4366
4367	4369	4378	4380	4384	4387	4388	4390	4391	4393	4403	4405	4409	4412	4416
4417	4419	4420	4423	4426	4429	4431	4446	4448	4459	4464	4465	4466	4468	4469
4470	4472	4476	4488	4493	4496	4500	4503	4505	4507	4508	4510	4528	4530	4540
4544	4553	4554	4557	4558	4563	4566	4568	4569	4571	4575	4576	4578	4589	4591
4596	4603	4604	4606	4607	4609	4610	4611	4613	4615	4625	4627	4633	4636	4639
4640	4643	4644	4646	4647	4648	4650	4657	4659	4664	4665	4667	4668	4669	4670
4672	4680	4682	4683	4687	4690	4692	4694	4705	4707	4711	4714	4715	4716	4717
4719	4728	4730	4736	4740	4741	4744	4745	4747	4748	4749	4750	4752	4760	4762
4765	4769	4770	4771	4774	4778	4783	4787	4790	4792	4799	4801	4804	4808	4811
4812	4814	4815	4816	4818	4836	4838	4870	4871	4879	4880	4883	4884	4888	4889
4892	4894	4898	4899	4904	4905	4906	4909	4912	4915	4917	4921	4923	4928	4931
4932	4933	4934	4936	4943	4945	4949	4954	4955	4957	4958	4959	4961	4963	4990
4992	5001	5002	5008	5009	5016	5017	5020	5021	5024	5025	5034	5035	5036	5037
5039	5047	5049	5052	5056	5057	5058	5062	5066	5070	5072	5079	5081	5084	5088
5089	5090	5095	5098	5102	5104	5113	5115	5118	5125	5128	5131	5132	5134	5135
5138	5139	5140	5144	5147	5150	5152	5160	5162	5164	5168	5171	5173	5175	5176
5179	5186	5189	5191	5199	5201	5208	5211	5214	5215	5217	5218	5221	5226	5229
5231	5240	5242	5246	5250	5253	5255	5257	5258	5261	5266	5270	5272	5280	5282
5287	5291	5294	5295	5297	5298	5301	5304	5307	5309	5322	5324	5327	5328	5332
5333	5335	5336	5339	5340	5343	5344	5346	5347	5348	5351	5355	5358	5360	5368
5370	5374	5378	5381	5382	5384	5385	5387	5391	5394	5396	5403	5405	5408	5412
5415	5417	5419	5420	5423	5429	5432	5434	5443	5445	5451	5455	5456	5457	5459
5460	5463	5467	5471	5473	5481	5483	5487	5491	5494	5495	5497	5498	5501	5504
5507	5509	5514	5516	5521	5524	5525	5526	5529	5533	5536	5538	5542	5544	5547
5550	5551	5552	5554	5556	5560	5562	5567	5570	5571	5572	5573	5575	5577	5579
5585	5586	5587	5589	5591	5599	5601	5605	5610	5612	5617	5620	5622	5623	5624
5626	5638	5640	5644	5649	5650	5654	5655	5658	5661	5665	5666	5669	5674	5676
5689	5690	5693	5694	5697	5700	5703	5704	5707	5709	5712	5714	5716	5717	5720
5722	5724	5726	5727	5729	5730	5732	5733	5741	5743	5745	5746	5748	5750	5762
5764	5775	5776	5777	5779	5781	5783	5784	5786	5788	5790	5792	5809	5816	5819
5820	5821	5825	5826	5827	5828	5829	5830	5834	5838	5841	5843	5844	5846	5852
5853	5855	5857	5930	6008	6076	6085	6089	6091	6092	6102	6103	6104	6105	6106
2441	2463	2465	2467	2473	2499	2515	2532	2623	2629	2630	2631	2639	2640	2646
2647	2650	2651	2652	2653	2661	2662	2667	2668	2672	2673	2675	2676	2677	2678
2686	2687	2693	2694	2698	2700	2701	2702	2703	2711	2712	2718	2719	2723	2725
2726	2727	2728	2736	2737	2743	2744	2748	2750	2751	2752	2756	2757	2758	2765
2766	2770	2771	2772	2773	2774	2775	2782	2783	2788	2789	2793	2794	2796	2797
2798	2799	2806	2807	2812	2813	2817	2819	2820	2821	2822	2829	2830	2835	2836
2840	2842	2843	2844	2845	2851	2852	2856	2857	2858	2859	2860	2861	2869	2870
2877	2878	2890	2891	2902	2903	2913	2914	2927	2928	2939	2940	2950	2951	2962
2963	2974	2975	2989	2990	3001	3002	3015	3016	3028	3029	3036	3037	3049	3050
3069	3070	3072	3073	3084	3085	3097	3098	3109	3110	3124	3125	3137	3138	3149
3150	3152	3153	3154	3170	3171	3179	3180	3182	3183	3186	3187	3198	3199	3201
3202	3206	3207	3217	3218	3220	3221	3223	3225	3226	3233	3234	3237	3238	3241
3242	3245	3246	3252	3253	3255	3256	3257	3258	3259	3263	3264	3267	3268	3271
3272	3277	3278	3281	3282	3283	3284	3285	3289	3290	3293	3294	3297	3298	3303
3304	3307	3308	3309	3310	3311	3315	3316	3319	3320	3323	3324	3329	3330	3333
3334	3335	3336	3337	3341	3342	3345	3346	3349	3350	3355	3356	3359	3360	3361
3362	3363	3367	3368	3371	3372	3375	3376	3381	3382	3385	3386	3387	3388	3389
3393	3394	3397	3398	3401	3402	3407	3408	3411	3412	3413	3414	3415	3420	3421

.IFF

3447	3448	3451	3452	3456	3457	3460	3461	3465	3466	3469	3470	3474	3475	3478
3479	3483	3484	3487	3488	3492	3493	3496	3497	3510	3520	3521	3522	3530	3531
3541	3548	3552	3559	3560	3561	3562	3563	3564	3567	3568	3569	3570	3571	3572
3585	3586	3591	3595	3596	3610	3611	3612	3613	3614	3615	3617	3618	3619	3620
3621	3622	3624	3625	3626	3627	3628	3629	3655	3656	3658	3673	3676	3677	3681
3684	3685	3686	3688	3689	3690	3695	3696	3697	3698	3701	3702	3703	3704	3707
3708	3709	3710	3711	3723	3724	3725	3726	3738	3739	3740	3741	3742	3743	3749
3756	3757	3758	3759	3760	3761	3762	3763	3778	3779	3784	3785	3786	3791	3792
3793	3799	3800	3804	3805	3806	3813	3814	3820	3821	3822	3829	3830	3836	3837
3838	3845	3846	3852	3853	3854	3860	3861	3863	3864	3865	3872	3873	3874	3881
3882	3883	3889	3890	3891	3892	3893	3909	3910	3912	3913	3925	3926	3927	3928
3931	3932	3933	3934	3940	3943	3948	3949	3950	3952	3953	3954	3967	3968	3970
3971	3974	3975	3976	3983	3984	3985	3986	3987	4005	4006	4010	4014	4015	4024
4025	4026	4027	4028	4029	4036	4037	4038	4041	4042	4043	4044	4045	4046	4047
4048	4055	4056	4057	4058	4061	4062	4063	4064	4079	4080	4083	4084	4088	4093
4094	4095	4097	4098	4099	4100	4101	4102	4103	4104	4114	4115	4119	4120	4123
4124	4125	4126	4127	4133	4134	4138	4142	4143	4144	4145	4146	4156	4157	4160
4161	4166	4167	4168	4169	4170	4171	4172	4175	4179	4181	4182	4183	4202	4203
4206	4207	4211	4212	4213	4214	4215	4216	4218	4219	4220	4221	4222	4223	4229
4230	4231	4232	4236	4237	4238	4239	4240	4248	4249	4250	4251	4255	4256	4257
4279	4280	4285	4286	4290	4291	4292	4293	4296	4297	4301	4303	4304	4309	4310
4318	4320	4321	4322	4323	4324	4339	4340	4344	4345	4351	4354	4355	4356	4357
4360	4361	4362	4365	4366	4367	4368	4369	4379	4380	4384	4385	4388	4391	4392
4393	4404	4405	4409	4410	4413	4416	4417	4418	4419	4420	4421	4424	4427	4429
4430	4431	4447	4448	4459	4460	4465	4466	4467	4468	4469	4470	4471	4472	4487
4488	4493	4494	4497	4500	4504	4505	4508	4509	4510	4529	4530	4540	4541	4545
4553	4554	4555	4557	4558	4559	4563	4567	4568	4569	4570	4571	4576	4577	4578
4590	4591	4596	4597	4603	4604	4605	4606	4607	4608	4610	4611	4612	4613	4614
4615	4626	4627	4633	4634	4637	4639	4640	4641	4643	4644	4645	4646	4647	4648
4649	4650	4658	4659	4664	4665	4666	4668	4669	4670	4671	4672	4681	4682	4683
4684	4687	4691	4692	4693	4694	4706	4707	4711	4712	4715	4716	4717	4718	4719
4729	4730	4736	4737	4740	4741	4742	4744	4745	4746	4748	4749	4750	4751	4752
4761	4762	4765	4766	4770	4771	4772	4775	4779	4784	4788	4790	4791	4792	4800
4801	4804	4805	4809	4811	4812	4813	4814	4815	4816	4817	4818	4837	4838	4871
4879	4880	4881	4883	4884	4885	4888	4889	4890	4892	4895	4898	4899	4900	4905
4906	4907	4910	4913	4915	4916	4917	4922	4923	4928	4929	4932	4933	4934	4935
4936	4944	4945	4949	4950	4954	4955	4956	4958	4959	4960	4961	4962	4963	4991
4992	5002	5008	5009	5010	5016	5017	5018	5020	5021	5022	5024	5025	5026	5035
5036	5037	5038	5039	5048	5049	5052	5053	5057	5058	5059	5063	5067	5070	5071
5072	5080	5081	5084	5085	5089	5090	5091	5096	5099	5102	5103	5104	5114	5115
5118	5119	5125	5129	5131	5132	5133	5134	5135	5136	5139	5140	5141	5145	5148
5150	5151	5152	5161	5162	5164	5165	5169	5171	5174	5175	5176	5177	5180	5187
5189	5190	5191	5200	5201	5208	5209	5212	5214	5215	5216	5217	5218	5219	5222
5227	5229	5230	5231	5241	5242	5246	5247	5251	5253	5256	5257	5258	5259	5262
5267	5270	5271	5272	5281	5282	5287	5288	5292	5294	5295	5296	5297	5298	5299
5302	5305	5307	5308	5309	5323	5324	5328	5332	5333	5334	5335	5336	5337	5339
5340	5341	5343	5344	5345	5347	5348	5349	5352	5356	5358	5359	5360	5369	5370
5374	5375	5379	5381	5382	5383	5384	5385	5386	5388	5392	5394	5395	5396	5404
5405	5408	5409	5413	5415	5418	5419	5420	5421	5424	5430	5432	5433	5434	5444
5445	5451	5456	5457	5458	5459	5460	5461	5464	5468	5471	5472	5473	5482	5483
5487	5492	5494	5495	5496	5497	5498	5499	5502	5505	5507	5508	5509	5515	5516
5521	5525	5526	5527	5530	5534	5536	5537	5538	5543	5544	5547	5551	5552	5553
5554	5555	5556	5561	5562	5567	5571	5572	5573	5574	5575	5578	5579	5586	5587
5588	5589	5590	5591	5600	5601	5606	5610	5613	5617	5621	5622	5623	5624	5625
5626	5639	5640	5644	5649	5650	5651	5655	5659	5662	5666	5667	5668	5675	5676
5689	5690	5691	5693	5694	5695	5697	5701	5704	5708	5709	5713	5714	5717	5718

	5721	5722	5725	5726	5728	5729	5731	5732	5734	5741	5744	5745	5746	5747	5748
	5749	5750	5763	5764	5775	5776	5777	5778	5779	5782	5783	5784	5785	5786	5789
	5790	5793	5820	5822	5829	5834	5838	5841	5853	5858	5931	6009	6077	6086	
.IFT	5810	5817													
.IFTF	5810	5817													
.IIF	2026	2031	2036	2435	2511	5813	5827	5828	5835	5853	5856	5930	6101	6102	6103
	6104	6105													
.IRP	2428	2511	2629	2647	2650	2651	2667	2668	2675	2676	2693	2694	2700	2701	2718
	2719	2725	2726	2743	2744	2750	2756	2771	2772	2773	2788	2789	2796	2797	2812
	2813	2819	2820	2835	2836	2842	2843	2857	2858	2859	2877	2878	3152	3179	3182
	3186	3187	3198	3201	3206	3217	3220	3225	3233	3237	3241	3245	3252	3256	3257
	3267	3268	3271	3272	3277	3278	3281	3282	3283	3293	3294	3297	3298	3303	3304
	3307	3308	3309	3319	3320	3323	3324	3329	3330	3333	3334	3335	3345	3346	3349
	3350	3355	3356	3359	3360	3361	3371	3372	3375	3376	3381	3382	3385	3386	3387
	3397	3398	3401	3402	3407	3408	3411	3412	3413	3447	3451	3456	3460	3465	3469
	3474	3478	3483	3487	3492	3496	3520	3559	3562	3567	3570	3595	3596	3610	3613
	3617	3620	3624	3627	3673	3677	3684	3688	3695	3698	3702	3707	3709	3738	3741
	3756	3759	3751	3784	3791	3799	3804	3813	3820	3829	3836	3845	3852	3860	3863
	3872	3881	3890	3891	3925	3928	3932	3940	3948	3952	3974	3983	3985	4024	4027
	4036	4041	4044	4046	4055	4058	4062	4093	4097	4100	4102	4123	4125	4142	4144
	4167	4170	4181	4211	4214	4219	4221	4236	4238	4255	4290	4293	4297	4320	4322
	4351	4355	4360	4365	4367	4388	4391	4416	4419	4429	4465	4468	4470	4500	4505
	4508	4553	4557	4563	4568	4571	4576	4603	4606	4610	4613	4639	4643	4646	4648
	4664	4668	4670	4687	4692	4715	4717	4740	4744	4748	4750	4770	4790	4811	4814
	4816	4879	4883	4888	4892	4898	4905	4915	4932	4934	4954	4958	4961	5008	5016
	5020	5024	5035	5037	5057	5070	5089	5102	5125	5131	5134	5139	5150	5171	5175
	5189	5214	5217	5229	5253	5257	5270	5294	5297	5307	5332	5335	5339	5343	5347
	5358	5361	5384	5394	5415	5419	5432	5456	5459	5471	5494	5497	5507	5525	5536
	5551	5554	5571	5573	5586	5589	5610	5617	5622	5624	5649	5655	5666	5689	5693
	5697	5704	5709	5714	5718	5722	5726	5729	5732	5741	5745	5748	5776	5779	5783
	5786	5790	5820	5834	6022	6062									
.LIST	1	2011	2427	2428	2435	2499	2501	2502	2503	2504	2505	2506	2507	2508	2531
	2538	2539	2540	2542	2543	2545	2547	2549	2551	2553	2556	2557	2559	2561	2562
	2563	2564	2566	2567	2569	2570	2572	2574	2575	2576	2578	2580	2582	2584	2588
	2589	2591	2593	2595	2597	2599	2601	2604	2605	2606	2608	2610	2612	2614	2616
	2617	2619	2620	2629	2640	2647	2650	2651	2662	2667	2668	2673	2675	2676	2687
	2693	2694	2699	2700	2701	2712	2718	2719	2724	2725	2726	2737	2743	2744	2749
	2750	2756	2766	2771	2772	2773	2783	2788	2789	2794	2796	2797	2807	2812	2813
	2818	2819	2820	2830	2835	2836	2841	2842	2843	2852	2857	2858	2859	2870	2877
	2878	2889	2901	2912	2926	2938	2949	2961	2973	2988	3000	3014	3027	3035	3048
	3068	3071	3083	3096	3108	3123	3136	3148	3152	3171	3178	3179	3181	3182	3186
	3187	3197	3198	3200	3201	3205	3206	3216	3217	3219	3220	3224	3225	3232	3233
	3236	3237	3240	3241	3244	3245	3251	3252	3255	3256	3257	3264	3267	3268	3271
	3272	3277	3278	3281	3282	3283	3290	3293	3294	3297	3298	3303	3304	3307	3308
	3309	3316	3319	3320	3323	3324	3329	3330	3333	3334	3335	3342	3345	3346	3349
	3350	3355	3356	3359	3360	3361	3368	3371	3372	3375	3376	3381	3382	3385	3386
	3387	3394	3397	3398	3401	3402	3407	3408	3411	3412	3413	3421	3446	3447	3450
	3451	3455	3456	3459	3460	3464	3465	3468	3469	3473	3474	3477	3478	3482	3483
	3486	3487	3491	3492	3495	3496	3520	3531	3540	3542	3559	3562	3567	3570	3586
	3590	3592	3595	3556	3610	3613	3617	3620	3624	3627	3656	3657	3659	3673	3677
	3684	3688	3695	3698	3702	3707	3709	3724	3725	3726	3738	3741	3756	3759	3761
	3779	3784	3790	3791	3798	3799	3803	3804	3812	3813	3819	3820	3828	3829	3835
	3836	3844	3845	3851	3852	3859	3860	3862	3863	3871	3872	3880	3881	3889	3890
	3891	3910	3912	3925	3928	3932	3940	3948	3952	3968	3970	3971	3974	3983	3985
	4006	4014	4015	4024	4027	4036	4041	4044	4046	4055	4058	4062	4080	4083	4084
	4093	4097	4100	4102	4115	4119	4120	4123	4125	4134	4137	4139	4142	4144	4157

4160	4161	4167	4170	4181	4203	4206	4207	4211	4214	4219	4221	4230	4231	4232	
4236	4238	4249	4250	4251	4255	4280	4285	4286	4290	4293	4297	4303	4304	4308	
4319	4320	4322	4340	4344	4345	4351	4355	4360	4365	4367	4380	4384	4385	4388	
4391	4405	4409	4410	4416	4419	4429	4448	4459	4460	4465	4468	4470	4488	4493	
4494	4500	4505	4508	4530	4540	4541	4553	4557	4563	4568	4571	4576	4591	4596	
4597	4603	4606	4610	4613	4627	4633	4634	4639	4643	4646	4648	4659	4664	4668	
4670	4682	4683	4684	4687	4692	4707	4711	4712	4715	4717	4730	4736	4737	4740	
4744	4748	4750	4762	4765	4766	4770	4790	4801	4804	4805	4811	4814	4816	4838	
4870	4872	4879	4883	4888	4892	4898	4905	4915	4923	4928	4929	4932	4934	4945	
4949	4950	4954	4958	4961	4992	5001	5002	5008	5016	5020	5024	5035	5037	5049	
5052	5053	5057	5070	5081	5084	5085	5089	5102	5115	5118	5119	5125	5131	5134	
5139	5150	5162	5164	5165	5171	5175	5189	5201	5208	5209	5214	5217	5229	5242	
5246	5247	5253	5257	5270	5282	5287	5288	5294	5297	5307	5324	5327	5329	5332	
5335	5339	5343	5347	5358	5370	5374	5375	5381	5384	5394	5405	5408	5409	5415	
5419	5432	5445	5451	5452	5456	5459	5471	5483	5487	5488	5494	5497	5507	5516	
5521	5522	5525	5536	5544	5547	5548	5551	5554	5562	5567	5568	5571	5573	5579	
5586	5589	5601	5610	5617	5622	5624	5640	5644	5645	5649	5655	5666	5676	5683	
5693	5697	5704	5709	5714	5718	5722	5726	5729	5732	5741	5745	5748	5764	5776	
5779	5783	5786	5790	5810	5817	5820	5835	5846	6091	6092	6101	6102	6103	6104	
6105	6106														
.MACRO	1	2466	2629	2646	2651	2672	2676	2701	2726	2756	2773	2793	2797	2820	
	2843	3151	3257	3283	3309	3335	3361	3387	3413	3520	3570	3627	3673	3676	
	3697	3761	3891	3927	3940	3952	3985	4046	4057	4102	4125	4144	4181	4221	
	4238	4255	4322	4351	4367	4388	4391	4429	4470	4500	4504	4508	4563	4570	
	4576	4613	4648	4670	4686	4717	4750	4790	4816	4892	4915	4934	4961	5037	
	5070	5102	5125	5150	5171	5189	5229	5253	5270	5307	5358	5415	5432	5471	
	5507	5536	5554	5573	5589	5610	5617	5624	5655	5666	5697	5704	5713	5717	
	5721	5725	5728	5731	5740	5748	5778	5785	5789						
.MCALL	2011	2427	5820												
.NLIST	1	2011	2427	2428	2435	2499	2501	2502	2503	2504	2505	2506	2507	2508	2531
	2538	2539	2540	2542	2543	2545	2547	2549	2551	2553	2556	2557	2559	2561	2562
	2563	2564	2566	2567	2569	2570	2572	2574	2575	2576	2578	2580	2582	2584	2588
	2589	2591	2593	2595	2597	2599	2601	2604	2605	2606	2608	2610	2612	2614	2616
	2617	2619	2620	2629	2640	2647	2650	2651	2662	2667	2668	2673	2675	2676	2687
	2693	2694	2699	2700	2701	2712	2718	2719	2724	2725	2726	2737	2743	2744	2749
	2750	2756	2766	2771	2772	2773	2783	2788	2789	2794	2796	2797	2807	2812	2813
	2818	2819	2820	2830	2835	2836	2841	2842	2843	2852	2857	2858	2859	2870	2877
	2878	2889	2901	2912	2926	2938	2949	2961	2973	2988	3000	3014	3027	3035	3048
	3068	3071	3083	3096	3108	3123	3136	3148	3152	3171	3178	3179	3181	3182	3186
	3187	3197	3198	3200	3201	3205	3206	3216	3217	3219	3220	3224	3225	3232	3233
	3236	3237	3240	3241	3244	3245	3251	3252	3255	3256	3257	3264	3267	3268	3271
	3272	3277	3278	3281	3282	3283	3290	3293	3294	3297	3298	3303	3304	3307	3308
	3309	3316	3319	3320	3323	3324	3329	3330	3333	3334	3335	3342	3345	3346	3349
	3350	3355	3356	3359	3360	3361	3368	3371	3372	3375	3376	3381	3382	3385	3386
	3387	3394	3397	3398	3401	3402	3407	3408	3411	3412	3413	3421	3446	3447	3450
	3451	3455	3456	3459	3460	3464	3465	3468	3469	3473	3474	3477	3478	3482	3483
	3486	3487	3491	3492	3495	3496	3520	3531	3540	3542	3559	3562	3567	3570	3586
	3590	3592	3595	3596	3610	3613	3617	3620	3624	3627	3656	3657	3659	3673	3677
	3684	3688	3695	3698	3702	3707	3709	3724	3725	3726	3738	3741	3756	3759	3761
	3779	3784	3790	3791	3798	3799	3803	3804	3812	3813	3819	3820	3828	3829	3835
	3836	3844	3845	3851	3852	3859	3860	3862	3863	3871	3872	3880	3881	3889	3890
	3891	3910	3912	3925	3928	3932	3940	3948	3952	3968	3970	3971	3974	3983	3985
	4006	4014	4015	4024	4027	4036	4041	4044	4046	4055	4058	4062	4080	4083	4084
	4093	4097	4100	4102	4115	4119	4120	4123	4125	4134	4137	4139	4142	4144	4157
	4160	4161	4167	4170	4181	4203	4206	4207	4211	4214	4219	4221	4230	4231	4232
	4236	4238	4249	4250	4251	4255	4280	4285	4286	4290	4293	4303	4304	4308	

	4319	4320	4322	4340	4344	4345	4351	4355	4360	4365	4367	4380	4384	4385	4388
	4391	4405	4409	4410	4416	4419	4429	4448	4453	4460	4465	4468	4470	4488	4493
	4494	4500	4505	4508	4530	4540	4541	4553	4557	4563	4568	4571	4576	4591	4596
	4597	4603	4606	4610	4613	4627	4633	4634	4639	4643	4646	4648	4659	4664	4668
	4670	4682	4683	4684	4687	4692	4707	4711	4712	4715	4717	4730	4736	4737	4740
	4744	4748	4750	4762	4765	4766	4770	4790	4801	4804	4805	4811	4814	4816	4838
	4870	4872	4879	4883	4888	4892	4898	4905	4915	4923	4928	4929	4932	4934	4945
	4949	4950	4954	4958	4961	4992	5001	5002	5008	5016	5020	5024	5035	5037	5049
	5052	5053	5057	5070	5081	5084	5085	5089	5102	5115	5118	5119	5125	5131	5134
	5139	5150	5162	5164	5165	5171	5175	5189	5201	5208	5209	5214	5217	5229	5242
	5246	5247	5253	5257	5270	5282	5287	5288	5294	5297	5307	5324	5327	5329	5332
	5335	5339	5343	5347	5358	5370	5374	5375	5381	5384	5394	5405	5408	5409	5415
	5419	5432	5445	5451	5452	5456	5459	5471	5483	5487	5488	5494	5497	5507	5516
	5521	5522	5525	5536	5544	5547	5548	5551	5554	5562	5567	5568	5571	5573	5579
	5586	5589	5601	5610	5617	5622	5624	5640	5644	5645	5649	5655	5666	5676	5689
	5693	5697	5704	5709	5714	5718	5722	5726	5729	5732	5741	5745	5748	5764	5776
	5779	5783	5786	5790	5810	5817	5820	5835	5846	6091	6092	6101	6102	6103	6104
.PAGE	2037	2466	2514												
.REM	1														
.REPT	2435	2501	2504												
.SBTTL	2038	2163	2174	2188	2337	2429	2436	2442	2468	2516	2629	2651	2676	2701	2726
	2756	2773	2797	2820	2843	2859	3152	3257	3283	3309	3335	3361	3387	3413	3519
	3520	3569	3570	3626	3627	3709	3761	3891	3952	3985	4046	4102	4125	4144	4181
	4221	4238	4255	4321	4322	4367	4391	4429	4470	4508	4576	4613	4648	4670	4691
	4692	4717	4750	4790	4816	4915	4934	4960	4961	5037	5070	5102	5150	5189	5229
	5269	5270	5307	5358	5394	5432	5470	5471	5507	5536	5553	5554	5573	5589	5624
	5666	5748	5823	5859	5932	6010	6078	6093							
.TITLE	2026														
.WORD	2435	2462	2464	2476	2479	2480	2481	2482	2485	2486	2487	2488	2489	2490	2499
	2501	2502	2503	2504	2505	2506	2507	2512	3911	3944	3969	3973	3981	4861	4864
	4867	4869	4876	4901	5838	5841	5927	6007	6106						

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

* ,DEKBAB.SEQ/SOL/CRF/PAGNUM/NL: TOC=DEKBAB.SML, DEKBAB.P11
 RUN-TIME: 78 100 15 SECONDS
 RUN-TIME RATIO: 280/194=1.4
 CORE USED: 36K (71 PAGES)

